

# **DIGITAL COMMUNICATION**

## **Second Edition**

**Edward A. Lee**

*University of California at Berkeley*

**David G. Messerschmitt**

*University of California at Berkeley*



**Kluwer Academic Publishers**  
*Boston/Dordrecht/London*

---

**Distributors for North America:**

Kluwer Academic Publishers  
101 Philip Drive  
Assinippi Park  
Norwell, Massachusetts 02061 USA

**Distributors for all other countries:**

Kluwer Academic Publishers Group  
Distribution Centre  
Post Office Box 322  
3300 AH Dordrecht, THE NETHERLANDS

---

**Library of Congress Cataloging-in-Publication Data**

Lee, Edward A., 1957-

Digital communication / Edward A. Lee and David G. Messerschmitt.

-- 2nd ed.

p. cm.

Includes bibliographical references and index.

ISBN 0-7923-9391-0 (acid-free paper)

1. Digital communications. I. Messerschmitt, David G.

II. Title.

TK5103.7.L44 1994

621.382--dc20

93-26197

CIP

---

**Copyright** © 1994 by Kluwer Academic Publishers. Fifth Printing 1999.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photo-copying, recording, or otherwise, without the prior written permission of the publisher, Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061.



# CONTENTS

---

PREFACE  
NOTES TO THE INSTRUCTOR

## PART I: THE BASICS

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	APPLICATIONS OF DIGITAL COMMUNICATION	2
1.2	DIGITAL vs. ANALOG COMMUNICATIONS	5
1.3	PLAN OF THE BOOK	7
1.4	FURTHER READING	8
<b>2</b>	<b>DETERMINISTIC SIGNAL PROCESSING</b>	<b>11</b>
2.1	SIGNALS	11
2.2	LTI SYSTEMS AND FOURIER TRANSFORMS	13
2.3	THE NYQUIST SAMPLING THEOREM	15
2.4	PASSBAND SIGNALS and MODULATION	17
2.5	Z TRANSFORMS AND RATIONAL TRANSFER FUNCTIONS	21
2.6	SIGNAL SPACE REPRESENTATIONS	31
2.7	FURTHER READING	39
2-A	SUMMARY OF FOURIER TRANSFORM PROPERTIES	39
2-B	SPECTRAL FACTORIZATION	41
<b>3</b>	<b>STOCHASTIC SIGNAL PROCESSING</b>	<b>48</b>
3.1	RANDOM VARIABLES	48
3.2	RANDOM PROCESSES	57
3.3	MARKOV CHAINS	68
3.4	THE POISSON PROCESS AND QUEUEING	75
3.5	FURTHER READING	85

3-A	POWER SPECTRUM OF A CYCLOSTATIONARY PROCESS	86
3-B	POWER SPECTRUM OF A MARKOV CHAIN	87
3-C	DERIVATION OF POISSON PROCESS	90
3-D	MOMENT GENERATING FUNCTION OF SHOT NOISE	91

## **4 LIMITS OF COMMUNICATION 97**

4.1	JUST ENOUGH INFORMATION ABOUT ENTROPY	99
4.2	CAPACITY OF DISCRETE-TIME CHANNELS	102
4.3	FURTHER READING	110
4-A	ASYMPTOTIC EQUIPARTITION THEOREM	110

## **5 PHYSICAL MEDIA AND CHANNELS 115**

5.1	COMPOSITE CHANNELS	116
5.2	TRANSMISSION LINES	119
5.3	OPTICAL FIBER	127
5.4	MICROWAVE RADIO	142
5.5	TELEPHONE CHANNELS	160
5.6	MAGNETIC RECORDING CHANNELS	167
5.7	FURTHER READING	171

# **PART II: MODULATION AND DETECTION**

## **6 MODULATION 178**

6.1	AN OVERVIEW OF BASIC PAM TECHNIQUES	179
6.2	PULSE SHAPES	187
6.3	BASEBAND PAM	191
6.4	PASSBAND PAM	199
6.5	ALPHABET DESIGN	213
6.6	THE MATCHED FILTER — ISOLATED PULSE CASE	224
6.7	SPREAD SPECTRUM	229
6.8	ORTHOGONAL MULTIPULSE MODULATION	230
6.9	COMBINED PAM AND MULTIPULSE MODULATION	249
6.10	OPTICAL FIBER RECEPTION	261
6.11	MAGNETIC RECORDING	262
6.12	FURTHER READING	263
6-A	MODULATING RANDOM PROCESSES	263
6-B	THE GENERALIZED NYQUIST CRITERION	266

## **7 SIGNAL and RECEIVER DESIGN 279**

- 7.1 SIGNAL MODEL 282
- 7.2 SPECIFIC MODULATION TECHNIQUES 286
- 7.3 PAM WITH INTERSYMBOL INTERFERENCE 294
- 7.4 BANDWIDTH and SIGNAL DIMENSIONALITY 304
- 7.5 FURTHER READING 307

## **8 NOISE 311**

- 8.1 COMPLEX-VALUED GAUSSIAN PROCESSES 311
- 8.2 FUNDAMENTAL RESULTS 316
- 8.3 PERFORMANCE of PAM 320
- 8.4 PERFORMANCE of MINIMUM-DISTANCE RECEIVERS 329
- 8.5 PAM with ISI 334
- 8.6 SPREAD SPECTRUM 337
- 8.7 CAPACITY AND MODULATION 344
- 8.8 QUANTUM NOISE in OPTICAL SYSTEMS 360
- 8.9 FURTHER READING 371

## **9 DETECTION 378**

- 9.1 DETECTION OF A SINGLE REAL-VALUED SYMBOL 380
- 9.2 DETECTION OF A SIGNAL VECTOR 385
- 9.3 KNOWN SIGNALS IN GAUSSIAN NOISE 390
- 9.4 OPTIMAL INCOHERENT DETECTION 402
- 9.5 OPTIMAL DETECTORS for PAM WITH ISI 406
- 9.6 SEQUENCE DETECTION: THE VITERBI ALGORITHM 409
- 9.7 SHOT NOISE SIGNAL WITH KNOWN INTENSITY 424
- 9.8 FURTHER READING 427
  - 9-A KARHUNEN-LOEVE EXPANSION 428
  - 9-B GENERAL ML AND MAP SEQUENCE DETECTORS 430
  - 9-C BIT ERROR PROBABILITY FOR SEQUENCE DETECTORS 432

## **10 EQUALIZATION 442**

- 10.1 OPTIMAL ZERO-FORCING EQUALIZATION 445
- 10.2 GENERALIZED EQUALIZATION METHODS 464
- 10.3 FRACTIONALLY SPACED EQUALIZER 482
- 10.4 TRANSVERSAL FILTER EQUALIZERS 486
- 10.5 ISI and CHANNEL CAPACITY 487
- 10.6 FURTHER READING 511
  - 10-A DFE ERROR PROPAGATION 511



## **PART IV: SYNCHRONIZATION**

### **15 PHASE-LOCKED LOOPS 700**

- 15.1 IDEAL CONTINUOUS-TIME PLL 702
- 15.2 DISCRETE-TIME PLLs 709
- 15.3 PHASE DETECTORS 713
- 15.4 VARIATIONS ON A THEME: VCOs 718
- 15.5 FURTHER READING 720

### **16 CARRIER RECOVERY 725**

- 16.1 DECISION-DIRECTED CARRIER RECOVERY 726
- 16.2 POWER OF N CARRIER RECOVERY 733
- 16.3 FURTHER READING 734

### **17 TIMING RECOVERY 737**

- 17.1 TIMING RECOVERY PERFORMANCE 739
- 17.2 SPECTRAL-LINE METHODS 741
- 17.3 MMSE TIMING RECOVERY AND APPROXIMATIONS 748
- 17.4 BAUD-RATE TIMING RECOVERY 754
- 17.5 ACCUMULATION OF TIMING JITTER 756
- 17.6 FURTHER READING 759
  - 17-A THE POISSON SUM FORMULA 759
  - 17-B DISCRETE-TIME DERIVATIVE 760

## **PART V: MULTIPLE ACCESS**

### **18 MULTIPLE ACCESS ALTERNATIVES 765**

- 18.1 MEDIUM TOPOLOGY FOR MULTIPLE ACCESS 767
- 18.2 MULTIPLE ACCESS BY TIME DIVISION 770
- 18.3 MULTIPLE ACCESS BY FREQUENCY DIVISION 787
- 18.4 MULTIPLE ACCESS BY CODE DIVISION 789
- 18.5 THE CELLULAR CONCEPT 792

## **19 ECHO CANCELLATION**

**797**

- 19.1 PRINCIPLE OF THE ECHO CANCELER 798
- 19.2 BASEBAND CHANNEL 801
- 19.3 PASSBAND CHANNEL 804
- 19.4 ADAPTATION 809
- 19.5 FAR-END ECHO 815
- 19.6 FURTHER READING 818
  - 19-A REAL-ERROR CANCELER CONVERGENCE 819

**EXERCISE SOLUTIONS 825**

**PERMUTED INDEX 865**

# PREFACE

This book concerns digital communication. Specifically, we treat the transport of bit streams from one geographical location to another over various physical media, such as wire pairs, coaxial cable, optical fiber, and radio waves. Further, we cover the multiple access and synchronization issues relevant to constructing communication networks that simultaneously transport bit streams from many users. The material in this book is thus directly relevant to the design of a multitude of digital communication systems, including for example local and metropolitan area data networks, voice and video telephony systems, digital CATV distribution, digital cellular and radio systems, the narrowband and broadband integrated services digital network (ISDN), computer communication systems, voiceband data modems, and satellite communication systems. We extract the common principles underlying these and other applications and present them in a unified framework.

This book is intended for *designers* and *would-be designers* of digital communication systems. To limit the scope to manageable proportions we have had to be selective in the topics covered and in the depth of coverage. In the case of advanced information, coding, and detection theory, for example, we have not tried to duplicate the in-depth coverage of many advanced textbooks, but rather have tried to cover those aspects directly relevant to the design of digital communication systems. For example, in our view it would be unfortunate to defer many of the insights of information theory to an advanced course on that topic, since the bounds they provide are directly relevant to the design of systems. Thus, we discuss the channel capacity results of information theory, as well as elementary derivations and justification of those results, without getting into the detail or rigor that the students will encounter in an advanced course on information theory. As another example, we restrict our coverage of detection theory to that portion especially relevant to the design of digital communication systems, such as the detection of known signals with additive Gaussian or quantum noise or after transmission over a binary-symmetric channel.

Our emphasis on topics important to designers leads us to more detailed treatment of some topics than is traditional in academic textbooks, for example in our coverage of synchronization. We devote several chapters to synchronization, including PLL's, timing recovery, and carrier recovery. Another example of a non-traditional topic is a description of the properties of the most important communication media, including radio, cable, and fiber. We then relate the modulation, detection, and coding techniques back to these properties. The book is also modern in its treatment of signal-space and trellis coding.

In the Second Edition, we have both tried to improve on some of the existing material, as well as add new material. Two major topics that we have concentrated on are the combination of coding with intersymbol interference, and fading channels and wireless communication. Both of these topics have been very active both in the literature and in commercial application since the First Edition. In terms of improving on the old development, we have completely rewritten Chapters 6-8 of the First Edition, turning them into Chapters 6-10 in the Second Edition. In addressing a perceived shortcoming of the First Edition, we have included solutions to the exercises at the end of the book. What follows is a chapter-by-chapter summary of the changes.

Chapter 2. We added a lot of new material on rational transfer functions, including spectral factorizations. Later, when equalization is covered, we concentrate on the rational case, since many concrete statements can be made with mathematical ease. Also added is a derivation of the basic modulation and demodulation, which is later used in Chapter 6.

Chapter 3. A small amount of material on innovations and linear prediction theory has been added. Linear prediction later plays a major role in Chapter 10 in the context of decision-feedback equalization.

Chapter 4. The capacity of the Gaussian vector channel is derived.

Chapter 5. The description of fading channels has been significantly upgraded, including derivation of standard channel models for narrowband and wideband Rayleigh fading channels. A short description of optical amplifiers has also been added.

Chapter 6. The probability of error derivations has been deferred to Chapter 8, and rather this chapter concentrates on signal-to-noise ratio derivations. Some topics of recent importance have been added, including spread spectrum (preceded by a derivation of the matched filter), code-division multiple access, and multicarrier modulation. A new generalized Nyquist criterion, that extends the Nyquist criterion to various types of orthogonal signaling, is derived and used to compare spectral efficiencies of the modulation techniques.

Chapter 7. This is a completely new chapter that follows a new approach for texts on digital communication. A simple signal space minimum-distance criterion is defined, and used to derive receiver structures for a variety of modulation techniques. By avoiding bringing in noise and optimality considerations, this chapter is able to derive all the standard receiver structures very quickly, and compare them. Surprisingly, this includes even the "whitened matched filter", normally derived from noise whitening considerations, but derived here solely from signal space considerations.

Chapter 8. This new chapter analyzes the probability of error of various receiver structures, including the receiver designs derived in Chapter 6 on the basis of intuitive and SNR considerations, as well as the minimum-distance receiver designs of Chapter 7. We begin by specifically treating complex-valued Gaussian noise, which is the subject of a lot of misconceptions in many textbooks. Many books, including our First Edition, seem to imply that complex Gaussian processes are characterized by their power spectrum, are stationary if they are wide-sense stationary, etc., statements that are all incorrect or incomplete. We attempt to clear this up by doing a more



complete treatment, and defining a desirable property called "circular symmetry" which guarantees the nice properties and which seems to be widely satisfied in digital communication systems. This chapter then proceeds to compare the modulation techniques by first deriving the capacity of an ideal white Gaussian noise channel, and then comparing the performance of each modulation technique to the capacity ideal. The particular approach used is to define a rate-normalized SNR and a "SNR gap to capacity", an approach used recently by Forney and Eyuboglu.

Chapter 9. This chapter, which is a revision of Chapter 7 of the First Edition, derives the optimal receiver structures based on probability of error criterion (ML and MAP). Basically, this chapter justifies the minimum-distance receiver designs already derived earlier in Chapter 7, and further extends them to colored noise. Rather than rely on whitening filter arguments as in the First Edition, we have done a full Karhunen-Loeve expansion approach. Chapter 10. This is a major revision of Chapter 8 of the First Edition, which now considers only intersymbol interference and equalization, rather than also doing receiver optimization as before. We start by finding optimal equalizer structures, which leads to matched-filter front ends. Then in contrast to the First Edition, as well as other texts, we remove the assumption of a matched filter front end, and re-optimize the equalizers. Our motivation here is based on two related facts: Matched filtering is often impossible to implement theoretically for non-minimum phase channels, and is seldom used in practice for unknown or time-varying channels. In our view, the traditional treatments of optimal equalization based on matched filtering obscure many practically-important issues, such as the quite different characteristics of minimum-phase and non-minimum phase channels. Our treatment maintains relative mathematical simplicity by concentrating on rational spectra. Also added to this chapter is substantial treatment of the capacity of channels with ISI (based on water pouring), and the effect of ISI on capacity. We also generalize the rate-normalized SNR to channels with ISI, and characterize thereby the "SNR gap to capacity" for different equalization techniques, establishing "Price's results" that the gap to capacity is often independent of the ISI when decision-feedback equalization is used.

In addition to those folks mentioned in the Preface to the First Edition, we owe a debt of gratitude to a number of additional friends and colleagues who assisted us by reading and commenting on selected Chapters in the second edition, or by providing useful reference material.

This book is suitable as a first-year graduate textbook, and should also be of interest to many professionals in industry. We have attempted to make the book more attractive to both audiences through the inclusion of many practical examples and a practical flavor in the choice of topics. In addition, we have increased the readability by relegating many of the more detailed derivations to appendices and exercises, both of which are included in the book. The inclusion of exercise solutions at the end of the book is new to the Second Edition. A solutions manual for the problems is available from the publisher.

We owe a debt of gratitude to a number of our friends and colleagues who have assisted us by reading and commenting on selected chapters. For the first edition,

these include Bob Aaron, Jeff Bier, John Barry, Graham Brand, Thomas Chen, Paul Frieburg, Biswa Ghosh, Vijay Madisetti, Teresa Meng, Sara Miller, Rhonda Richter, Ruth Schaefer, Gil Sih, Mehmet Soyuer, Aram Thomasian, Ho-Ping Tseng, and Greg Uehara, as well as the students in EECS 225 in the Spring of 1987. The assistance of several anonymous reviewers in making sure all the essential topics are covered is gratefully acknowledged. The diligent and artistic effort of Pei Ku in generating many of the figures is appreciated. For the Second Edition, additional colleagues provided proofreading assistance, including Shuvra Bhattacharyya, Shih-fu Chang, Wan-the Chang, Soonhoi Ha, Paul Haskell, Chih-Tsung Huang, Joseph Kahn, William Li, Jean-Paul Linnartz, Vijay Madisetti, Praveen Murthy, Sun-Inn Shih, S. Sriram, and Louis Yun, as well as the students in EECS 224 in the Fall of 1992. Special thanks goes to G.David Forney, Jr. and John Barry, who each devoted many hours to proofreading the manuscript and suggesting added material or improved derivations. While many of these colleagues have pointed out many errors and omissions, any remaining errors are of course the full responsibility of the authors.

We hope the result is a readable and useful book, and always appreciate comments and suggestions from the readers.

*Edward A. Lee*

*David G. Messerschmitt*

Berkeley, California

June 12, 1993

# NOTES TO THE INSTRUCTOR

This book can be used as a textbook for advanced undergraduates, or for a first course in digital communication for graduate students. We presume a working knowledge of transforms, linear systems, and random processes, and review these topics in chapters 2 and 3 at a depth suitable only for establishing notation. This treatment also serves to delimit the background assumed in the remainder of the book. We include a more detailed treatment of basic topics important to digital communication but which may not be familiar to a first-year graduate student, including signal space (chapter 2), Markov chains and their analysis (chapter 3), Poisson processes and shot noise (chapter 3), the basic boundaries of communication from information theory (chapter 4), and maximum likelihood detection and the Viterbi algorithm (chapter 9). These treatments are self-contained and assume only the basic background mentioned earlier. These basic topics can be covered at the beginning of the course, or can wait until the first time they are used. Our own preference is the latter, since the immediate application of the techniques serves as useful reinforcement.

The core of book is the treatment of communications media (chapter 5), modulation (chapter 6), detection and equalization (chapters 7 through 11), coding (chapters 12 through 14), and synchronization (chapters 15 through 17). These topics are covered in considerable depth. After completing a course based on this book, students should be highly motivated to take advanced courses in information theory, algebraic coding, detection and estimation theory, and communication networks, and will have a prior appreciation of the utility of these topics.

There is sufficient material in this book for two semesters of instruction, although it can easily be used for a single-semester course by selectively covering topics. At Berkeley we use this book for a one-semester graduate course that has as prerequisites undergraduate courses in systems and transforms and probability and random processes. We do not presume any prior exposure to signal space, Markov chains, or the Poisson process. In this course we rely on the students to review Chapters 1 through 4 themselves, and we cover Chapters 5 through 10 and Chapter 13 and 14 in lecture. Chapter 11 is skipped because adaptive filtering techniques are covered in another signal processing course.

# 1

---

## INTRODUCTION

---

*But let your communication be, Yea, yea; Nay, nay:  
for whatever is more than these, cometh of evil.*

— The Gospel According to St. Matthew (5:37)

Digital transmission of information has sufficiently overwhelming advantages that it increasingly dominates communication systems. In computer-to-computer communication, the information to be transported is inherently digital, so digital transmission is the only practical alternative. But computer communication is still a small fraction of all digital communications. A much larger fraction is devoted to transmitting inherently analog signals, particularly speech and images. Such signals can be (and traditionally have been) transmitted in analog form. Why would they be transmitted digitally? Not so long ago, digital transmission of voice and video was considered wasteful of bandwidth, and the cost of converting from analog to digital and back was of concern. But four things have happened to change all that:

- The encoding of analog signals in digital form has benefited from advances in compression algorithms, which reduce dramatically the bit rate required to represent a voice or video signal with high subjective quality.

- Signal processing and coding techniques have dramatically increased the bit rate that can be supported through a given physical channel.
- Integrated circuits have greatly reduced the cost of realizing complex signal processing and coding functions inherent in digital transmission.
- Optical fiber has reduced the cost of transmitting high bit rates over long distances.

The result of these developments is a complete turnabout in thinking. Today the greatest impetus for digital transmission is often the *reduced* bandwidth, or equivalently the greater overall system capacity that can be achieved with digital transmission. For example, both cellular telephone and cable television are in the process of converting to digital transmission, in part on the basis of the greater system capacity. In fact, today virtually all communication is either already digital, in the process of being converted to digital, or under consideration for conversion.

An essential element in this digital revolution is the transmission of a higher and higher bit rates over a given physical transmission medium. That is the subject of this book.

## 1.1. APPLICATIONS OF DIGITAL COMMUNICATION

Digital communication is used for signals that are inherently analog and continuous-time, such as speech and images, and signals that are inherently digital, such as text files. The demands placed on a communication system are different in each case. Furthermore, modern communication networks provide a mixture of services, and hence must take into account the demands of each.

### 1.1.1. Digital Transmission of Speech and Video

It is common in modern practice to convert analog continuous-time signals to digital form for transmission. A technique commonly used for transmission of speech and video is *pulse code modulation (PCM)*, shown in Figure 1-1. The continuous-time speech or video signal is first *sampled* at a rate  $f_s$  Hz (or samples/sec), which must be greater than twice the highest frequency component in the signal (see Section

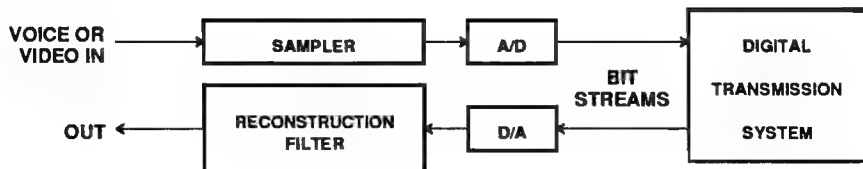


Figure 1-1. A digital transmission system used for PCM transmission.

2.3). Often this sampling operation is preceded by a lowpass filter, not shown, which ensures that the signal is properly bandlimited. Each sample is then converted to an  $n$ -bit binary word by an analog-to-digital converter (A/D). The output is a bit stream with a bit rate of  $nf_s$  bits per sec (often written bps or b/s). The bit stream is then transmitted over a digital transmission system, and the voice or video signal is reconstructed with a digital-to-analog converter (D/A) and a reconstruction low-pass filter. PCM was patented in France by Reeves in the late 1930's, and was first used during World War II.

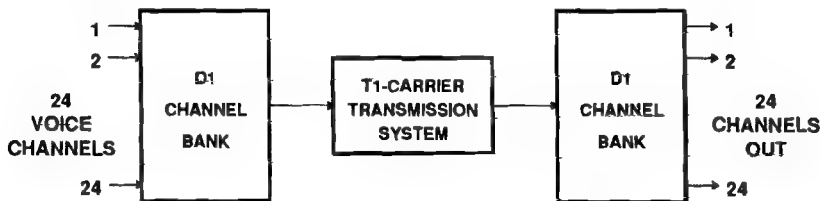
#### Example 1-1.

The first commercial PCM system was *T1-carrier*, shown in Figure 1-2. The design was completed at Bell Laboratories in 1962. It is still widely used for relatively short-distance transmission between central offices in metropolitan areas. The T1-carrier system transmits a bit stream at 1.544 Mb/s ( $1.544 \times 10^6$  b/s) over two wire-pairs, one for each direction of transmission. Of this capacity, 1.536 Mb/s is available for the transmission of 24 voice channels with the rest used for supervisory functions (Chapter 18). Each voice channel is sampled 8000 times per second and is quantized to eight bits per sample, for a total bit rate of 64 kb/s ( $64 \times 10^3$  b/s) per voice channel. The channel bank also *multiplexes* or combines the 24 bit streams together into the 1.536 Mb/s stream. □

The internal implementation of such a digital transmission system can be intricate, and is the subject of this book. However, from the viewpoint of voice and video transmission, the digital transmission system is simple in that it can be fully characterized by only four parameters:

- the *bit rate*,
- the *propagation and processing delay*,
- the *probability of error* (Chapter 8), which indicates how likely the bits arriving at the destination are to differ from the transmitted bits, and
- the *timing jitter* in the arriving bit stream (Chapter 17).

Both bit errors and timing jitter can cause some degradation in the quality of the recovered voice or video signal, and excessive delay can impair a conversation, so the designer of the digital communication system must control these impairments. They are, however, far less complicated than the types of distortion commonly encountered



**Figure 1-2.** The T1-carrier transmission system with associated D1 channel bank for sampling, A/D conversion, and multiplexing.

in analog transmission.

Another application of digital communication techniques is *storage systems* using *magnetic* or *optical media*. In this case the objective is not transmission "from here to there" but rather "from now to then." These media have unique impairments, different from those in transmission media, but many of the same basic techniques apply.

### 1.1.2. Computer Communication

The data generated by computers for transmission to other computers or terminals does not require the conversion shown in Figure 1-1. For computer communication it is common to use the term *data transmission* rather than *digital transmission*. But we should emphasize that the same transmission system can be used for PCM transmission and for data transmission. In fact, a major goal of many current design activities is to combine voice, video, and data within an *integrated* transmission environment (Chapter 22). For this reason, in this book we lump the two categories under the common heading *digital communication*.

A digital communication system used for data transmission is characterized by its bit rate, delay, error rate, and jitter, just like a system used for PCM transmission. The relative importance of these impairments is different, however. Data transmission is usually much more sensitive to bit errors than speech and video signals, but on the other hand is usually unaffected by jitter. Also, data often requires only sporadic rather than continuous communication. For example, a user sitting at a terminal may desire that the individual characters be transmitted at 9.6 kb/s, but the infrequently generated characters may require an actual average bit rate of only 50 b/s. We can characterize this property more formally as a big difference between peak and average bit rates. Special techniques have therefore been developed for data (Chapter 21). These often introduce additional impairments like delay variation with time and lost bits.

### 1.1.3. Telecommunication Networks

Practical applications require more than a single point-to-point digital communication link. A *digital communication network* simultaneously connects many users to one another, and often includes *switching*, which allows the users to initiate connections to specific other users. For historical reasons, networks designed for these different purposes are often assigned different terminology: a *telephone* or *telephony network* is designed to handle PCM transmission, and a *data network* or *computer network* is designed to transmit data. However, the terminology is rapidly becoming obsolete through the introduction of networks designed for all sources. Such networks we will call *telecommunication networks*. A good example is the *integrated services digital network (ISDN)* that is currently being deployed worldwide.

Telecommunication networks are often classified according to geographical extent. *Local area networks* are designed to communicate at very high bit rates over a small geographical area (on the order of one km in extent). Most existing local area networks are designed primarily for computer communication. *Metropolitan area networks* are currently being conceptualized as a wider area counterpart to the local

area network, and can cover an area extending approximately 50 kilometers using optical fiber. *Wide area networks*, encompassing as much as a country or the world, use a combination of cable, terrestrial microwave radio, underwater cable and fiber, and satellite, to cover large distances. The world-wide telephony network is an example of such a network. Another example is the *internet*, which is used primarily for computer communications.

Networks that transport multiple bit streams also include some form of switching. Switching enables reconfiguration of point-to-point connections in the network, and usually takes one of two basic forms — *circuit switching* or *packet switching* (Chapter 18). In circuit switching, the network connects a constant-rate bit stream to the destination for a relatively long period of time (of the order of minutes or longer). This mode arose in the context of voice networks, where the circuit is formed for the duration of one telephone call. In packet switching, the data is encapsulated into relatively short bundles of bits called packets, and a destination address is appended. The packet can then be routed through the network. One advantage of packet switching is that the bit stream between source and destination can have a variable bit rate — this is accomplished by generating only the needed packets. This leads to greater efficiency for sources of data that have a large ratio of peak to average bit rate.

## 1.2. DIGITAL vs. ANALOG COMMUNICATIONS

For data communication, there is no practical alternative to digital transmission. For voice and video signals, however, there are important advantages and disadvantages to digital transmission.

### Interface Abstraction

The relatively simple characterization of a digital communication system is an important advantage over analog communication, where there are many more ways in which a transmission can be degraded. For example, the signal-to-noise ratio may be poor, or the signal may suffer second or third order intermodulation distortion, or crosstalk from one user to another, or the system may clip the input signal. By contrast a digital communication system has only three parameters: bit rate, probability of error, and timing jitter. The impairments of the physical medium, which may be quite severe, are largely hidden from the user. Also hidden are the implementation details of the digital communication system itself. This property we call the *interface abstraction* of the transmission system. The power of this abstraction was perhaps first appreciated by Claude Shannon in his classic 1948 paper which started the field of information theory (Chapter 4). He showed that theoretically there is nothing lost by defining the interface between the signal to be transmitted and the transmission system to be a bit stream, regardless of whether the signal is analog or digital.



## The Regenerative Effect

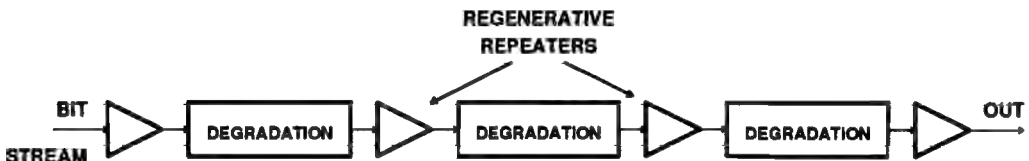
Consider the problem of transporting a bit stream over a long distance. The degradation of an uninterrupted physical medium, such as a cable or optical fiber, may be unacceptable. The digital solution is to place *regenerative repeaters* at periodic intervals in the physical medium, as shown in Figure 1-3. Each of these repeaters includes a receiver, which detects the bit stream just as is done at the eventual destination, and a re-transmitter similar to the one at the origination. In the absence of transmission bit errors, the bit stream that is regenerated in this repeater is *identical* to the one that was originally transmitted. Any effects due to noise and distortion on the physical medium have been *completely removed* by the regenerative repeater (except insofar as they cause occasional bit errors). Contrast this to analog transmission, where the equivalent repeaters consist basically of amplifiers, and the noise and distortion accumulates as the distance increases.

In practice, the digital communication system will introduce some errors in the detection of the bit stream. When the probability of error at each repeater is low, the total probability of error for  $m$  repeaters is approximately  $m$  times the probability of error for a single repeater. Assuming a maximum number of repeaters, we can derive from an overall probability of error objective a performance requirement for a single repeater. We can then meet that objective by adjusting the design of the repeater, the design of the signal, and the spacing between repeaters (closer spacing will result in a lower error rate for a given physical medium of transmission).

The same regenerative effect applies to the storage of signals. Consider for example the high quality possible with digital audio. Part of the reason for this is that each time the audio signal is copied onto a new medium (for example from a magnetic tape to a laser disk), the signal is regenerated, and degradations peculiar to the original medium are removed.

## Economics

We have seen that digital communication has some powerful technical advantages. What are the relative *economics* of digital and analog communication? There is a price to be paid for the technical advantages. Without getting into detail, a qualitative comparison can be made as follows.



**Figure 1-3.** A chain of regenerative repeaters reduces the effect of cascaded degradations by regenerating the digital signal at intermediate points in the transmission.

- The multiplexing and switching of digital signals is much lower in cost than for analog signals. This is particularly true for multiplexing, because frequency-division multiplexing of analog signals requires complicated filters (Chapter 18).
- Some economical media, such as optical fiber and laser disks, are better suited to digital transmission than analog.
- Digital communication of analog waveforms such as voice and video requires an additional step of sampling and analog-to-digital conversion. The cost of this conversion was initially an impediment to the widespread use of digital communication, but with integrated circuit technology this cost is rapidly becoming insignificant, particularly for voiceband signals.
- Regenerative repeaters for digital communication are considerably more complicated than their analog counterparts (which are just amplifiers). However, the capacity of these systems is so large that this added cost is insignificant to individual users.
- With modern compression and transmission technology (the latter being the subject of this book), PCM transmission of analog signals can be accomplished with less bandwidth than analog transmission of the same signal. This characteristic is critical in radio transmission, because the radio spectrum is in short supply. It can also have important economic advantages for other media, such as the telephone subscriber loop and coaxial cable television.
- Digital communication raises complicated synchronization issues (Chapters 15-19) that are largely avoided in analog communication.

The bottom line is that it took a while, about 20 years, for digital communication to almost completely supplant its analog competitors, but that revolution is now nearly complete. This is a result of a combination of economic factors, technological advances, and demands for new services.

## 1.3. PLAN OF THE BOOK

This book concentrates on the techniques that are used to design a digital communication system starting with any of the common physical media. Our concern is thus with how to get a bit stream from one location to another, and not so much with how this bit stream is used. In the particular context of a computer network, this aspect of the system is called the *physical layer*. We also address the problems of multiple bit streams sharing a common medium, called *multiple-access*.

In Chapters 2-4 some basics required for the understanding of later material are covered. Many readers will have a prior background in many of these basics, in which case only a superficial reading to pick up notation is required. We have also covered some basic topics with which the reader may not be so familiar. These include spectral factorization of rational transfer functions (Chapter 2), signal space (Chapter 2), Markov chains and Poisson processes (Chapter 3), and information theoretic bounds (Chapter 4). The characteristics of the physical media commonly encountered are covered in Chapter 5.

Chapters 6-14 cover the theory of modulation, detection, and coding that is necessary to understand how a single bit stream is transported over a physical medium. The need for this theory arises because all physical media are analog and continuous-time in nature. It is ironic that much of the design of a digital communication system is inevitably related to the analog and continuous-time nature of the medium, even though this is not evident at the abstracted interface to the user.

The design of a digital communication system or network raises many difficult *synchronization* issues that are covered in Chapters 15-17. Often a large part of the effort in the design of a digital communication system involves phase-locked loops, timing, and carrier recovery.

A complete telecommunication network requires that many bit streams originating with many different users be transported simultaneously while sharing facilities and media. This leads to the important topic of *multiple access* of more than one user to a single physical medium for transmission. This is covered in Chapters 18-19.

## 1.4. FURTHER READING

There are a number of excellent books on digital communication. While these books have a somewhat different emphasis from this one, they provide very useful supplementary material. The books by Roden [1], Benedetto, Biglieri, and Castellani [2], and Gitlin, Hayes, and Weinstein [3] cover similar material to this one, perhaps with a bit less practical emphasis. The books by Blahut [4] and Bingham [5] are valued for their practical orientation. Two texts provide additional detail on topics in this book: the recent book by Proakis [6] is an excellent treatise on applied information theory and advanced topics such as coding, spread spectrum, and multipath channels; the book by Viterbi and Omura [7] gives a detailed treatment of source and channel coding as applied to digital communication, as does Biglieri, Divsalar, McLane, and Simon [8]. An excellent treatment of the statistical communication theory as applied to digital communication is given by Schwartz [9]. On the topics of modulation, equalization, and coding the book by Lucky, Salz, and Weldon is somewhat dated but still recommended reading [10]. The same applies to the book by Wozencraft and Jacobs, which emphasizes principles of detection [11]. Books by Keiser and Strange [12] and Bellamy [13] give broad coverage of digital transmission at a descriptive level. Practical details of digital transmission can be found in a book published by AT&T Technologies [14], in the book by Bylanski and Ingram [15], and for the particular case of PCM encoding, in the book by Cattermole [16]. A couple of books expand on our brief description of digital switching, including McDonald [17] and Pearce [18]. For the information theory background that gives a solid theoretical foundation for digital communication, the books by Gallager [19], Cover and Thomas [20], Blahut [21], and McEliece [22] are recommended. Schwartz [23] and Bertsekas and Gallager [24] are recommended comprehensive texts on computer networks. There are also many elementary texts that cover both digital and analog communication, as well as the basic systems, transforms, and random process theory. Simulation techniques for communication systems are covered comprehensively in Jeruchim,

Balaban, Shanmugan [25].

## PROBLEMS

- 1-1. For an A/D converter, define a signal-to-error ratio as the signal power divided by the quantization error power, expressed in dB. A uniform quantizer, which has equally-spaced thresholds, has two parameters: the number of bits  $n$  and the step size  $\Delta$ .
  - (a) If we were to increase  $n$  by one, to  $n+1$ , for the same input signal, what would be the appropriate change to  $\Delta$ ?
  - (b) Without doing a detailed analysis, what do you suppose would be the effect on signal-to-error ratio of increasing from  $n$  to  $n+1$  bits/sample?
  - (c) What effect will this same change have on the bit rate?
  - (d) Using the prior results, what is the *form* of the relationship between signal-to-error ratio and the bit rate? (You may have unknown constants in your equation.)
- 1-2. An analog signal is transmitted using a PCM system. Discuss qualitatively the effects of bit errors on the recovered analog signal.
- 1-3. Discuss qualitatively the sources of delay that you would expect in a PCM system.
- 1-4. Suppose you have a source of data that outputs a bit stream with a bit rate that varies with time, but also has a peak or maximum bit rate. Describe qualitatively how you might transmit this bit stream over a link that provides a constant bit rate.

## REFERENCES

1. M. S. Roden, *Digital And Data Communication Systems*, Prentice-Hall, Englewood Cliffs, N.J. (1982).
2. S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1987).
3. R. D. Gitlin, J. F. Hayes, and S. B. Weinstein, *Data Communications Principles*, Plenum Press, New York and London (1992).
4. R. E. Blahut, "Digital Transmission of Information," *Addison-Wesley*, (1990).
5. J. A. C. Bingham, *The Theory and Practice of Modem Design*, John Wiley & Sons, New York (1988).
6. J. G. Proakis, *Digital Communications, Second Edition*, McGraw-Hill Book Co., New York (1989).
7. A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill (1979).
8. E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications*, Macmillan, New York (1991).
9. M. Schwartz, *Information Transmission, Modulation, and Noise*, McGraw-Hill, New York (1980).
10. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., *Principles of Data Communication*, McGraw-Hill Book Co., New York (1968).

11. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York (1965).
12. B. E. Keiser and E. Strange, *Digital Telephony and Network Integration*, Van Nostrand Reinhold, New York (1985).
13. J. Bellamy, *Digital Telephony*, John Wiley, New York (1982).
14. Bell Laboratories Members of Technical Staff, *Transmission Systems for Communications*, Western Electric Co., Winston-Salem N.C. (1970).
15. P. Bylanski and D. G. W. Ingram, *Digital Transmission Systems*, Peter Peregrinus Ltd., Stevenage England (1976).
16. K. W. Cattermole, *Principles of Pulse Code Modulation*, Iliffe Books Ltd., London England (1969).
17. J. C. McDonald, *Fundamentals of Digital Switching*, Plenum Press, New York (1983).
18. J. G. Pearce, *Telecommunications Switching*, Plenum, New York (1981).
19. R. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, Inc., New York (1968).
20. T. M. Cover and J. A. Thomas, "Elements of Information Theory," Wiley, (1991).
21. R. E. Blahut, "Principles and Practice of Information Theory," Addison-Wesley, (1987).
22. R. J. McEliece, *The Theory of Information and Coding*, Addison Wesley Pub. Co. (1977).
23. M. Schwartz, *Telecommunication Networks: Protocols, Modeling, and Analysis*, Addison-Wesley, Reading, Mass. (1987).
24. D. Bertsekas and R. Gallager, "Data Networks," Prentice-Hall, (1987).
25. M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems*, Plenum Press, New York (1992).

# 2

---

## DETERMINISTIC SIGNAL PROCESSING

---

In this chapter we review some basic concepts in order to establish the notation used in the remainder of the book. In addition, we cover in more detail several specific topics that some readers may not be familiar with, including complex signals and systems, the convergence of bilateral Z-transforms, and signal space geometry. The latter allows simple geometric interpretation of many signal processing operations, and demonstrates relationships among many seemingly disparate topics.

### 2.1. SIGNALS

A *continuous-time signal* is a function  $x(t)$  of the real valued variable  $t$ , usually denoting time. A *discrete-time signal* is a sequence  $\{x_k\}$ , where  $k$  usually indexes a discrete progression in time. Throughout this book we will see systems containing both continuous-time and discrete-time signals. Often a discrete-time signal results from *sampling* a continuous-time signal; this is written  $x_k = x(kT)$ , where  $T$  is the *sampling interval*, and  $2\pi/T$  is the sampling frequency, in radians per second. The sampling operation can be represented as

$$x_k = x(kT) = \int_{-\infty}^{\infty} x(\tau) \delta(\tau - kT) d\tau, \quad (2.1)$$

where  $\delta(\tau)$  is the *Dirac delta function* or *continuous-time impulse*. The discrete-time

signal  $x_k$  has a continuous-time *pulse amplitude modulation (PAM)* representation

$$\hat{x}(t) = \sum_{k=-\infty}^{\infty} x_k \delta(t - kT), \quad (2.2)$$

in terms of impulses.

A continuous-time signal can be constructed from a discrete-time signal as represented symbolically in Figure 2-1. A discrete-time input to a continuous-time system implies first the generation of the continuous-time impulse train in (2.2), and then its application to a continuous-time filter  $F(j\omega)$ , yielding

$$y(t) = \sum_{k=-\infty}^{\infty} x_k f(t - kT). \quad (2.3)$$

### 2.1.1. Complex-Valued Signals

In digital communication systems, complex-valued signals are often a convenient mathematical representation for a pair of real-valued signals. A complex-valued signal consists of a *real* signal and an *imaginary* signal, which may be visualized as two voltages induced across two resistors or two sequences of numbers.

#### Example 2-1.

A complex-valued signal we encounter frequently is the complex exponential,

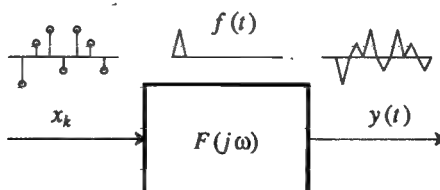
$$\begin{aligned} x_k &= e^{-j\omega kT} = \cos(\omega kT) - j \sin(\omega kT), \\ x(t) &= e^{-j\omega t} = \cos(\omega t) - j \sin(\omega t). \end{aligned} \quad (2.4)$$

We consistently use  $j$  to represent  $\sqrt{-1}$ .  $\square$

Complex-valued signals are processed just as real-valued signals are, except that the rules of complex arithmetic are followed.

#### Exercise 2-1.

Draw diagrams specifying the addition and multiplication of two complex-valued continuous-time signals in terms of real-valued additions and multiplications.  $\square$



**Figure 2-1.** Construction of a continuous-time signal from a discrete-time signal. When we show a discrete-time input to a continuous-time system, we imply first the generation of the impulse train in (2.2). An example is shown above the system.

The real part of the signal  $x(t)$  is written  $\text{Re}\{x(t)\}$  and the imaginary part  $\text{Im}\{x(t)\}$ . In addition, we write the complex conjugate of a signal  $x(t)$  as  $x^*(t)$ , and the squared modulus as  $|x(t)|^2$ . We don't use any special notation to distinguish real-valued from complex-valued signals because it will generally be clear from context. Complex signals will often be represented in block diagrams using double lines, as shown in Figure 2-2b.

### 2.1.2. Energy and Average Power

The energy of a signal  $x(t)$  or  $\{x_k\}$  is defined to be

$$\int_{-\infty}^{\infty} |x(t)|^2 dt, \quad \sum_{k=-\infty}^{\infty} |x_k|^2. \quad (2.5)$$

The average power is

$$\lim_{\tau \rightarrow \infty} \frac{1}{2\tau} \int_{-\tau}^{+\tau} |x(t)|^2 dt, \quad \lim_{K \rightarrow \infty} \frac{1}{(2K+1)T} \sum_{k=-K}^{+K} |x_k|^2. \quad (2.6)$$

## 2.2. LTI SYSTEMS AND FOURIER TRANSFORMS

The Fourier transform is valuable in the analysis of modulation systems and linear time-invariant systems. For the convenience of the reader, the properties of both discrete and continuous-time Fourier transforms are summarized in appendix 2-A. In this section we establish notation and review a few basic facts.

### 2.2.1. Linear Time Invariant (LTI) Systems

If a system *linear* and *time invariant* (LTI), then it is characterized by its pulse response  $h_k$  (for a discrete-time system) or  $h(t)$  (for a continuous-time system). The output of the LTI system can be expressed in terms of the input and impulse response as a convolution; for the discrete-time case,

$$y_k = x_k * h_k = \sum_{m=-\infty}^{\infty} x_m h_{k-m}, \quad (2.7)$$

and the continuous-time case,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau) d\tau. \quad (2.8)$$

An LTI system is *real* if its impulse response is real-valued, and *complex* if its impulse response is complex-valued. A complex system can be represented, using the rules of complex arithmetic, as a set of four real systems, as shown in Figure 2-2.

#### Exercise 2-2.

Show that if a complex system has a real-valued input it can be implemented using two real



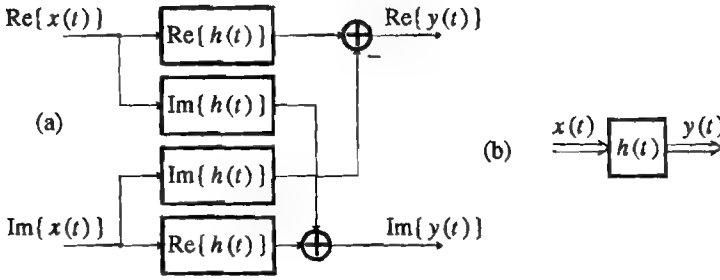


Figure 2-2. A complex-valued LTI system with a complex-valued input and output.

systems and sketch the configuration. Show that the same is true if a real system has a complex-valued input, and again sketch the configuration.  $\square$

### Exercise 2-3.

The notion of linearity extends to complex LTI systems. Demonstrate that if the four real systems required to implement a complex system are linear, then the resulting complex system is linear. It follows immediately that real-valued LTI systems are linear with respect to complex-valued inputs.  $\square$

## 2.2.2. The Fourier Transform

The Fourier transform pair for a continuous-time signal  $x(t)$  is

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt, \quad x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega)e^{+j\omega t} d\omega \quad (2.9)$$

while the discrete-time Fourier transform (DTFT) pair for  $x_k$  is

$$X(e^{j\omega T}) = \sum_{k=-\infty}^{\infty} x_k e^{-j\omega k T}, \quad x_k = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} X(e^{j\omega T}) e^{j\omega k T} d\omega. \quad (2.10)$$

The notation  $X(e^{j\omega T})$  deserves some explanation.  $X(e^{j\omega T})$  is the Z-transform  $X(z)$ , defined as

$$X(z) = \sum_{k=-\infty}^{\infty} x_k z^{-k}, \quad (2.11)$$

evaluated at  $z = e^{j\omega T}$ . Furthermore, the argument of the function,  $e^{j\omega T}$ , is periodic in  $\omega$ , emphasizing that the DTFT itself is periodic in  $\omega$  with period equal to the sampling rate  $2\pi/T$ . The  $j$  in  $X(j\omega)$  comes from the observation that  $X(j\omega)$  is the Laplace transform  $X(s)$  evaluated at  $s = j\omega$ .

If  $h(t)$  is the impulse response of a continuous-time system, then the Laplace transform  $H(s)$  is called the *transfer function*, and the Fourier transform  $H(j\omega)$  is called the *frequency response*. Correspondingly, for a discrete-time impulse response  $h_k$ , the transfer function is  $H(z)$  and the frequency response is  $H(e^{j\omega T})$ . Discrete-

time and continuous-time systems will often be distinguished only by the form of the argument of their transfer function or frequency response.

#### Exercise 2-4.

Starting with the convolution, show that the Fourier transform of the output of an LTI system is

$$Y(j\omega) = H(j\omega)X(j\omega), \quad Y(e^{j\omega T}) = H(e^{j\omega T})X(e^{j\omega T}) \quad (2.12)$$

for the continuous-time and discrete-time cases, where  $X(j\omega)$  and  $X(e^{j\omega T})$  are the Fourier transforms of the input signals.  $\square$

The magnitude of the frequency response  $|H(j\omega)|$  or  $|H(e^{j\omega T})|$  is called the *magnitude response*. The argument of the frequency response  $\arg(H(j\omega))$  or  $\arg(H(e^{j\omega T}))$  is called the *phase response*. The reason for these terms is explored in Problem 2-2.

A fundamental result allows us to analyze any system with a combination of continuous-time and discrete-time signals.

#### Exercise 2-5.

Given the definition (2.2) of a continuous-time PAM signal  $\hat{x}(t)$  derived from a discrete-time signal  $x_k$ , show that for all  $\omega$

$$\hat{X}(j\omega) = X(e^{j\omega T}). \quad (2.13)$$

In words, the Fourier transform of a PAM representation of a discrete-time signal is equal to the DTFT of the discrete time signal for all  $\omega$ .  $\square$

## 2.3. THE NYQUIST SAMPLING THEOREM

Suppose that we sample a continuous-time signal  $x(t)$  to get

$$x_k = x(kT). \quad (2.14)$$

From (2.2) we obtain

$$\hat{x}(t) = x(t) \sum_{m=-\infty}^{\infty} \delta(t - mT). \quad (2.15)$$

Multiplication in the time-domain corresponds to convolution in the frequency domain, so

$$\begin{aligned}
 \hat{X}(j\omega) &= \frac{1}{2\pi} \left[ X(j\omega) \right] * \left[ \frac{2\pi}{T} \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi}{T}m) \right] \\
 &= \frac{1}{T} \int_{-\infty}^{\infty} X(j\Omega) \sum_{m=-\infty}^{\infty} \delta(\omega - \Omega - \frac{2\pi m}{T}) d\Omega \\
 &= \frac{1}{T} \sum_{m=-\infty}^{\infty} X[j(\omega - \frac{2\pi m}{T})].
 \end{aligned} \tag{2.16}$$

Combining this with (2.13) we get the very important relation

$$X(e^{j\omega T}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} X[j(\omega - \frac{2\pi m}{T})]. \tag{2.17}$$

This fundamental *sampling theorem* relates the signals  $x(t)$  and  $x_k$  in the frequency domain. Systems with both discrete and continuous-time signals can now be handled easily.

#### Exercise 2-6.

Use (2.17) to show that the frequency response of a completely discrete-time system equivalent to that in Figure 2-3 is

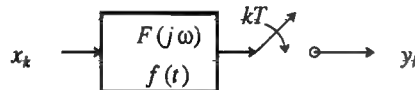
$$F(e^{j\omega T}) = \frac{1}{T} \sum_m F[j(\omega + m \frac{2\pi}{T})]. \tag{2.18}$$

□

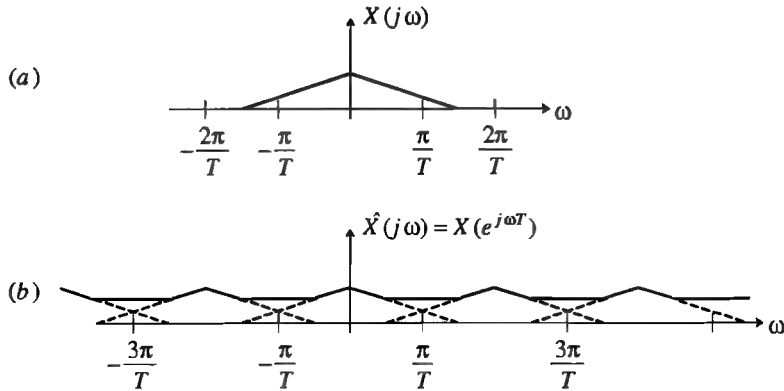
Notice that in (2.17) a component of  $X(j\omega)$  at any  $\omega = \omega_0$  is indistinguishable in the sampled version from a component at  $\omega = \omega_0 + 2\pi m/T$  for any integer  $m$ . This phenomenon is called *aliasing*.

#### Example 2-2.

Given a signal  $x(t)$  with Fourier transform  $X(j\omega)$  shown in Figure 2-4a, the Fourier transform of the sampled signal  $\hat{X}(j\omega) = X(e^{j\omega T})$  is shown in Figure 2-4b. The overlap evident in Figure 2-4b, called *aliasing distortion*, makes it very difficult to recover  $x(t)$  from its samples. □



**Figure 2-3.** A discrete-time system using a continuous-time filter.



**Figure 2-4.** The Fourier transform of a continuous-time signal (a) and its sampled version (b), where the sample rate is  $2\pi/T$ .

### Exercise 2-7.

(Nyquist sampling theorem.) Show that, from (2.17), a continuous-time signal can be reconstructed from its samples if it is sampled at a rate at least twice its highest frequency component. More precisely, if a signal  $x(t)$  with Fourier transform  $X(j\omega)$  is sampled at frequency  $2\pi/T$  (radians per second), then  $x(t)$  can be reconstructed from the samples if  $X(j\omega) = 0$  for all  $|\omega| > \pi/T$ .  $\square$

The sampling theorem gives a sufficient but not necessary condition for reconstructing a signal from its samples. In the absence of aliasing distortion, a lowpass signal can be reconstructed from its samples using an ideal low pass filter with cutoff frequency  $\pi/T$ ,

$$x(t) = \hat{x}(t) * \left[ \frac{\sin(\pi t/T)}{\pi t/T} \right] = \sum_{m=-\infty}^{\infty} x_m \frac{\sin[\pi(t-mT)/T]}{\pi(t-mT)/T}. \quad (2.19)$$

## 2.4. PASSBAND SIGNALS and MODULATION

Passband signals are fundamentally important for digital communication over channels, such as radio, where the signal spectrum must be confined to a narrow band of frequencies. In this section we will first define a useful building block called a phase splitter, then develop a complex baseband representation for any passband signal, and finally describe several useful modulation techniques for translating the frequency spectrum of a signal.

### 2.4.1. Phase Splitter and Analytic Signal

A *phase splitter* is a filter with impulse response  $\phi(t)$  and transfer function  $\Phi(j\omega)$ , where

$$\Phi(j\omega) = \begin{cases} 1, & \omega \geq 0 \\ 0, & \omega < 0 \end{cases} \quad (2.20)$$

The filter passes only positive frequencies, and rejects negative frequencies. Clearly, since  $\Phi(j\omega)$  does not display complex-conjugate symmetry,  $\phi(t)$  is a complex-valued impulse response. Regardless of the input to a phase splitter, the output must have only positive frequency components. A signal with only positive frequency components is called an *analytic signal*. Obviously, any analytic signal is complex-valued in the time domain.

Closely related to the phase splitter is the *Hilbert transform*, a filter with transfer function

$$H(j\omega) = -j \operatorname{sgn}(\omega) \quad (2.21)$$

It has a real-valued impulse response, since its transfer function has complex-conjugate symmetry. A Hilbert transform does not modify the amplitude spectrum of the input, but does give a  $-\pi$  phase shift at all frequencies. If the input to  $H(j\omega)$  is  $x(t)$ , then the output, the Hilbert transform of  $x(t)$ , is denoted by  $\hat{x}(t)$ .

#### Exercise 2-8.

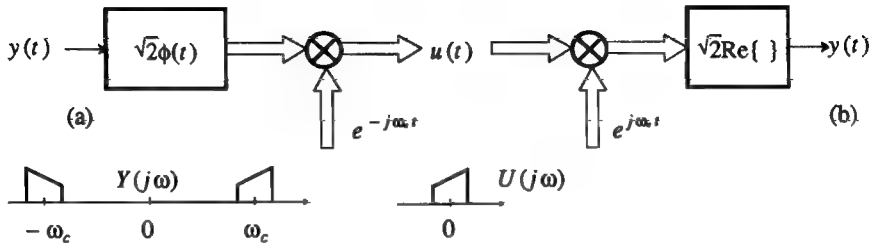
If the real-valued input to a phase splitter is  $x(t)$ , then show that the output is  $\frac{1}{2}\{x(t) + j\hat{x}(t)\}$ . Thus, the real part of the output is half the input, and the imaginary part is half the Hilbert transform of the input.  $\square$

The phase splitter and Hilbert transform filter both have a discontinuity in either amplitude or phase at d.c. They are therefore very difficult to implement at baseband frequencies. However, in many applications, we will apply a phase splitter to a passband signal, which makes the implementation much easier because the transfer function in the region of d.c. actually does not matter!

### 2.4.2. Complex Baseband Representation of Passband Signals

Suppose  $y(t)$  is a real-valued passband signal that happens to have a spectrum centered at  $\omega = \omega_c$ . We can develop a representation of  $y(t)$  in terms of a complex-valued baseband signal  $u(t)$ ; that is, a signal with its spectrum concentrated at d.c. Consider the system shown in Figure 2-5a. Since  $y(t)$  is real-valued, it has a spectrum concentrated at  $-\omega_c$  as well as  $\omega_c$ . If we pass  $y(t)$  first through a phase splitter, then the output analytic signal is missing the negative frequency terms. The remaining positive frequency terms can be shifted to d.c. by multiplying by a complex exponential  $e^{-j\omega_c t}$ , yielding the complex baseband representation  $u(t)$ . We add the strange-looking factor of  $\sqrt{2}$  for a good reason. Mathematically, the complex baseband signal can be represented as

$$u(t) = \frac{1}{\sqrt{2}}(y(t) + j\hat{y}(t))e^{-j\omega_c t} \quad (2.22)$$



**Figure 2-5.** Derivation of the complex baseband representation  $u(t)$  from a passband signal  $y(t)$ . (a) Obtaining  $u(t)$  from  $y(t)$ . (b) Recovering  $y(t)$  from  $u(t)$ . Also shown are typical spectra of the two signals, where  $U(j\omega)$  is a replica of the positive-frequency components of  $Y(j\omega)$  shifted to d.c.

### Exercise 2-9.

Show that  $u(t)$  has the same energy as  $y(t)$ , because of the factor of  $\sqrt{2}$ . This equal-energy property is important when we deal with noise and signal-to-noise ratios (Chapter 6).  $\square$

As shown in Figure 2-5b, the original passband signal can be recovered from the complex baseband representation through the equation

$$y(t) = \sqrt{2} \cdot \text{Re}\{ u(t)e^{j\omega_c t} \}. \quad (2.23)$$

This can also be easily verified by substituting (2.22) into (2.23). (2.23) is called the *canonical representation* of a passband signal in terms of a complex baseband signal. Any real-valued passband signal can be represented in this canonical form, where  $u(t)$  can be determined from (2.22) or Figure 2-5a.

### 2.4.3. Modulation

It is often useful to shift the spectrum of a signal, a process known as *modulation*.

#### Example 2-3.

A telephone channel passes only frequencies in the range from about 300Hz to about 3300Hz. Any signal transmitted over such a channel must be bandlimited in the same range, or it will not get through intact. Similarly, commercial broadcast AM radio occupies electromagnetic frequencies from 550kHz to 1.6MHz. An audio signal is limited to below 20kHz. Modulation is necessary to translate an audio signal into a frequency band suitable for AM transmission.  $\square$

Actually, modulation is the opposite of the canonical representation; rather than deriving an equivalent baseband representation of a passband signal, modulation generates an equivalent passband representation of a baseband signal. The canonical representation teaches us that a real-valued passband signal (necessary to transmit over a physical medium, Chapter 5) corresponds in general to a complex-valued baseband signal. Thus, assume that the baseband signal  $u(t)$  is complex-valued, and generate modulated signal (2.23). The resulting modulator and demodulator are shown in Figure 2-6, which is actually the same as Figure 2-5 reversed. Again, the factor  $\sqrt{2}$  ensures

that the modulated signal  $y(t)$  has the same energy as the baseband signal  $u(t)$ .

This representation of modulation is very general. All of the commonly used modulation techniques can be represented in this form. These techniques are distinguished by how they map an information-bearing signal into the complex baseband signal  $u(t)$ . We will illustrate this with three important modulation techniques: AM-DSB, AM-SSB, and QAM.

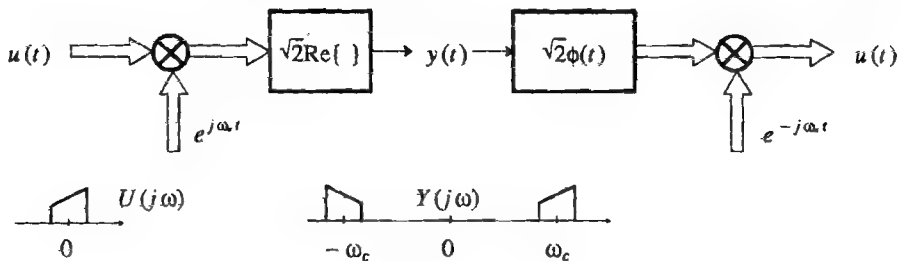
In *amplitude modulation double sideband (AM-DSB)*, a real-valued information-bearing signal  $a(t)$  is mapped into a passband signal by letting  $u(t) = a(t)$ . The baseband signal  $u(t)$  is therefore real-valued, and has a spectrum that is symmetric about the carrier frequency. The passband signal is represented mathematically as

$$y(t) = \sqrt{2} \cdot \text{Re}\{ a(t)e^{j\omega_c t} \} = \sqrt{2} \cdot a(t) \cdot \cos(\omega_c t) \quad (2.24)$$

The passband signal is complex-conjugate symmetric about the carrier frequency  $\omega_c$ , because the baseband signal is real-valued.

In *amplitude modulation single sideband (AM-SSB)* we again have a real-valued information-bearing signal  $a(t)$ , but we let the complex baseband signal be an analytic signal obtained by passing  $a(t)$  through a phase splitter,  $u(t) = \frac{1}{2}(a(t) + j\hat{a}(t))$ . Since the complex baseband signal is analytic, the passband signal has only upper-sideband frequency components above the carrier frequency. The advantage of AM-SSB over AM-DSB is that for the same  $a(t)$ , the bandwidth of the AM-SSB passband signal is half that of the AM-DSB signal, basically because the upper and lower sidebands of AM-DSB are complex-conjugate duplicates of one another. A disadvantage of AM-SSB is the baseband phase splitter, which can be difficult to realize because of the phase discontinuity at d.c. unless the baseband signal should happen to be missing frequencies near d.c. (as is true of telephone speech).

The third modulation technique is *quadrature amplitude modulation (QAM)*. In this case, we have two real-valued information-bearing signals  $a(t)$  and  $b(t)$ , and simultaneously modulate them by letting  $u(t) = a(t) + jb(t)$ . The complex baseband signal is neither analytic, nor has complex-conjugate symmetry about d.c.; the passband signal has in general both upper and lower sidebands and no particular



**Figure 2-6.** A modulator turns a complex baseband signal  $u(t)$  into a real-valued passband signal  $y(t)$ , by simply reversing Figure 2-5.

symmetry about the carrier frequency. The term "QAM" arises from the representation

$$y(t) = \sqrt{2} \cdot \text{Re}\{ (a(t) + jb(t))e^{j\omega_c t} \} = \sqrt{2} \cdot a(t) \cdot \cos(\omega_c t) - \sqrt{2} \cdot b(t) \cdot \sin(\omega_c t) \quad (2.25)$$

In other words, a QAM signal consists of two independently modulated carrier signals with a  $\pi/2$  relative phase shift. For the same baseband signal bandwidth, QAM requires the same passband bandwidth as AM-DSB, which is double that required for AM-SSB; however, it transmits two real-valued information-bearing signals rather than one. Thus, it offers the same spectral efficiency as AM-SSB, but without the requirement for the difficult-to-implement baseband phase splitter. QAM and similar modulation techniques have therefore become the most widely used for digital communication (Chapter 6).

## 2.5. Z TRANSFORMS AND RATIONAL TRANSFER FUNCTIONS

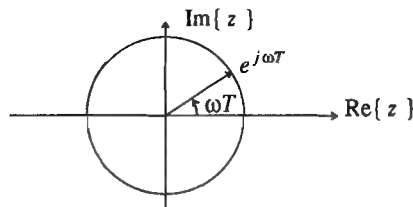
The Z transform, which is closely related to the DTFT, is particularly useful in the study of rational transfer functions. The Z transform is defined as

$$H(z) = \sum_{k=-\infty}^{\infty} h_k z^{-k}, \quad (2.26)$$

where  $z$  is a complex variable. As pointed out before, the DTFT is the Z transform evaluated at  $z = e^{j\omega T}$ , or on the unit circle in the  $z$  plane, as shown in Figure 2-7. This justifies the notation  $H(e^{j\omega T})$  for a DTFT. When  $\{h_k\}$  is the impulse response of a discrete-time LTI system, then  $H(z)$  is called the transfer function of the system. The transfer function on the unit circle is called the frequency response.

### 2.5.1. One Sided Sequences

A *causal* sequence  $\{h_k\}$  has  $h_k = 0$  for  $k < 0$ . An *anti-causal* sequence has  $h_k = 0$  for  $k > 0$ . A *right-sided* sequence is one for which, for some  $K$ ,  $h_k = 0$  for  $k < K$ . A *left-sided* sequence correspondingly has  $h_k = 0$  for  $k > K$  for some  $K$ . When  $h_k$  is the impulse response of an LTI system, that system is obviously causal



**Figure 2-7.** The Fourier transform of a discrete-time signal is the Z transform evaluated on the unit circle.



(anti-causal) if the impulse response is right-sided (left-sided) for  $K = 0$ . While physically realizable real-time LTI systems are causal, we will frequently find it useful to model systems as non-causal.

#### Example 2-4.

Assume a communication channel has the impulse response shown in Figure 2-8a. We can think of this channel as having a *flat propagation delay* of  $M$  samples plus the non-causal response  $\{h_k\}$  as shown in Figure 2-8b. Often the flat delay will not be an essential feature of the channel, in which case we ignore it.  $\square$

#### Example 2-5.

Suppose we come up with a non-causal filter  $H(z)$  in a theoretical development. This need not concern us too much, since such a filter can be approximated by a causal filter  $G(z)$  together with an additional flat delay  $z^{-M}$ . This extra flat delay which did not arise in the theoretical development will often not harm the system.  $\square$

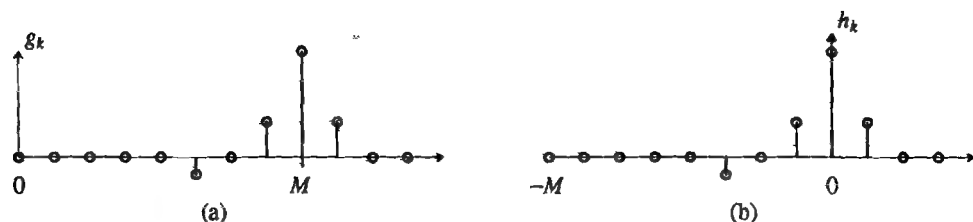
A particularly important class of causal or anti-causal sequences have a unity-valued sample at time zero ( $h_0 = 1$ ). These sequences are said to be *monic*. A similar terminology is used for polynomials, which are monic if the constant term is unity. It is easy to see from (2.26) that  $H(z)$  is the Z transform of a causal and monic sequence if and only if  $H(\infty) = 1$ . Similarly, it is anti-causal and monic if and only if  $H(0) = 1$ . When a sequence  $\{h_k\}$  is monic, then  $H(z)$ , as a polynomial in  $z$ , is also monic.

The *region of convergence* (ROC) of the Z transform is the region of the  $z$  plane where the series in (2.26) is absolutely summable,

$$\sum_{k=-\infty}^{\infty} |h_k z^{-k}| < \infty. \quad (2.27)$$

Note that for any  $z \in \text{ROC}$ ,  $|H(z)| < \infty$ , because

$$|H(z)| \leq \sum_{k=-\infty}^{\infty} |h_k z^{-k}| < \infty. \quad (2.28)$$



**Figure 2-8.** Illustration of the usefulness of a non-causal channel model. (a) Actual channel impulse response. (b) A non-causal version of the impulse response where the flat propagation delay is ignored.

Since the Fourier transform is the Z transform evaluated on the unit circle, for signals with a Fourier transform the ROC includes the unit circle.

A *bounded-input bounded-output (BIBO) stable* system has the property that any bounded input sequence with  $|x_k| < L$  produces a bounded output sequence with  $|y_k| < K$ . We will often use the term "stable" to denote "BIBO stable".

**Exercise 2-10.**

Show that a system with impulse response  $\{h_k\}$  is BIBO stable if and only if

$$S = \sum_{k=-\infty}^{\infty} |h_k| < \infty. \quad (2.29)$$

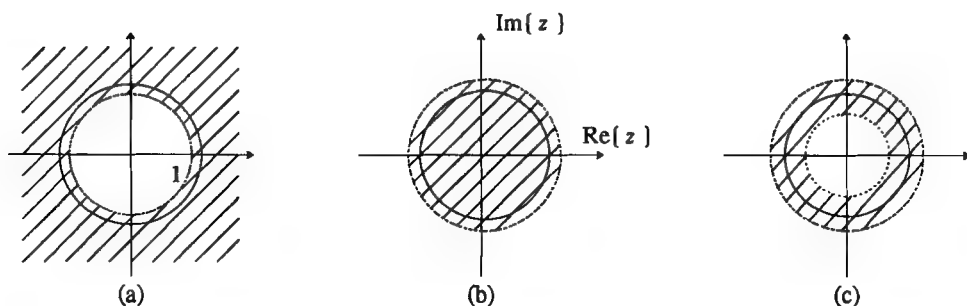
□

A consequence of this is that a system is BIBO stable if and only if the ROC includes the unit circle. To see this, note that (2.27) can be rewritten

$$\sum_{k=-\infty}^{\infty} |h_k| |z|^{-k} < \infty. \quad (2.30)$$

On the unit circle,  $|z| = 1$ , so this sum equals  $S$  in (2.29). By analogy with BIBO stable systems, a *sequence*  $\{h_k\}$  (not necessarily an impulse response) is said to be stable if it is absolutely summable, as in (2.29).

It is evident from (2.30) that the ROC depends only on  $|z|$ ; that is, it is of the form of an *annulus* or doughnut-shaped region. For a causal sequence, the ROC will be of the form  $|z| > R$  for some constant  $R$ . In words, the ROC will be the region outside a circle of radius  $R$ . If the sequence is also stable, then  $R < 1$ , as shown in Figure 2-9a. To see this, note that for a causal sequence, the summation in (2.30) becomes



**Figure 2-9.** The ROC of the Z transform of a stable sequence must include the unit circle. Three cases of stable sequences are illustrated: (a) A right-sided, (b) left-sided, and (c) two-sided sequence. The ROC includes  $|z| = \infty$  in (a) if the sequence is causal. It includes  $z = 0$  in (b) if the sequence is anti-causal.

$$\sum_{k=0}^{\infty} |h_k| |z|^{-k} < \infty. \quad (2.31)$$

All the terms in the summation are positive powers of  $|z|^{-1}$ , and hence get smaller as  $|z|$  gets larger. Thus, if absolute convergence occurs for some  $|z_1| > R$ , it will occur for all  $z$  such that  $|z| \geq |z_1|$ .

If the sequence is right-sided but not causal, (2.30) becomes

$$\sum_{k=K}^{\infty} |h_k| |z|^{-k} < \infty. \quad (2.32)$$

for some  $K < 0$ . The positive powers of  $|z|$  do not converge at  $z = \infty$ , but do converge at all other  $z$ . Thus, the ROC cannot include  $|z| = \infty$ , and should be written  $R < |z| < \infty$ . Similar results apply to left-sided sequences.

### Exercise 2-11.

- Show that the ROC of a left-sided stable sequence is of the form  $0 < |z| < R$  for  $R > 1$ .
- Show that a left-sided sequence is anti-causal if and only if its ROC includes the origin,  $0 \leq |z| < R$ , as shown in Figure 2-9b.  $\square$

To summarize, a right-sided sequence has an ROC consisting of the region outside a circle. That region includes  $|z| = \infty$  if and only if the sequence is causal. A left-sided sequence has an ROC consisting of the inside of a circle. That region includes  $z = 0$  if and only if the sequence is anti-causal. In all cases, the ROC includes the unit circle if and only if the sequence is stable.

## 2.5.2. Rational Transfer Functions

A rational transfer function can be written in any of the forms

$$H(z) = z^r \cdot \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} = A \cdot z^r \cdot \frac{\prod_{k=1}^M (1 - c_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} = A \cdot z^m \cdot \frac{\prod_{k=1}^M (z - c_k)}{\prod_{k=1}^N (z - d_k)}, \quad (2.33)$$

where  $A = b_0 / a_0$  and  $m = N - M + r$ . Notice that in the middle form, the numerator and denominator polynomials are both monic. The ratio of two such monic polynomials is also monic (carry out the long division to verify this).

The system has  $M$  zeros (roots of the numerator) at  $c_k$ ,  $1 \leq k \leq M$ , and  $N$  poles (roots of the denominator) at  $d_k$ ,  $1 \leq k \leq N$ . The factor  $z^m$  represents merely an advance or delay in the impulse response. If  $m > 0$  this factor introduces  $m$  zeros at the origin and  $m$  poles at  $|z| = \infty$  (conversely for  $m < 0$ ). If  $h_k$  is real valued, then  $H(z)$  in (2.33) has real-valued coefficients, and the zeros and poles are always either real valued or come in complex-conjugate pairs (Problem 2-20).

Including poles and zeros at  $z = 0$  and  $|z| = \infty$ , every rational transfer function has the same number of poles and zeros. This will be illustrated by two examples.

**Example 2-6.**

The causal FIR transfer function  $H(z) = 1 - 0.5z^{-1}$  has one zero at  $z = 1/2$  and one pole at  $z = 0$ . The only possible ROC is  $|z| > 0$ , which is a degenerate case of Figure 2-9a.  $\square$

**Example 2-7.**

The anti-causal FIR transfer function  $H(z) = 1 - 0.5z$  has one zero at  $z = 2$  and one pole at  $|z| = \infty$ . The only possible ROC is  $|z| < \infty$ , which is a degenerate case of Figure 2-9b.  $\square$

The ROC cannot include any of the poles, since  $H(z)$  is unbounded there. Moreover, for rational transfer functions, the ROC is bordered by poles. Referring to Figure 2-9, for a causal and stable  $H(z)$ , all poles must be inside the unit circle. For an anti-causal and stable  $H(z)$ , all poles must be outside the unit circle. No stable  $H(z)$  can have poles on the unit circle, although it can certainly have zeros on the unit circle.

**Exercise 2-12.**

LTI systems that can actually be implemented with computational hardware can be represented by linear constant-coefficient difference equations with zero initial conditions. Show that the system represented by

$$y_k = \frac{1}{a_0} \left( \sum_{l=0}^M b_l x_{k-l} - \sum_{l=1}^N a_l y_{k-l} \right) \quad (2.34)$$

has transfer function given by (2.33) with  $r = 0$ .  $\square$

When the denominator in (2.33) is unity ( $N = 0$ ), the system has a *finite impulse response (FIR)*, otherwise it has an *infinite impulse response (IIR)*. FIR systems are always stable, and are often a good approximation to physical systems. They can have poles only at  $z = 0$  and  $|z| = \infty$ , and the ROC therefore includes the entire  $z$  plane with the possible exception of  $z = 0$  and  $|z| = \infty$ . If an FIR system is causal, it has no poles at  $|z| = \infty$ . If it is anti-causal, it has no poles at  $z = 0$ .

**Example 2-8.**

Physical channels, such as a coaxial cable (Chapter 5), usually do not have, strictly speaking, a rational transfer function. However, they can be adequately approximated by a rational transfer function. Often the simplest approximation is FIR, obtained by simply truncating the actual impulse response for sufficiently large  $M$ . Alternatively, it may be possible to approximate the response with fewer parameters using an IIR transfer function.  $\square$

**2.5.3. Allpass Transfer Functions**

An *allpass* transfer function is any transfer function where the magnitude frequency response is unity for all  $\omega$ ,

$$|H_{\text{allpass}}(e^{j\omega T})| = 1. \quad (2.35)$$

This can be written as

$$H_{\text{allpass}}(e^{j\omega T})H_{\text{allpass}}^*(e^{j\omega T}) = 1. \quad (2.36)$$

Applying the inverse DTFT, we get

$$h_k * h_{-k}^* = \delta_k, \quad (2.37)$$

where  $h_k$  is the impulse response of  $H_{\text{allpass}}(e^{j\omega T})$ . Taking Z transforms we see that

$$H_{\text{allpass}}(z)H_{\text{allpass}}^*(1/z^*) = 1. \quad (2.38)$$

For rational Z transforms, (2.38) implies that every pole of  $H_{\text{allpass}}(z)$  is cancelled by a zero of  $H_{\text{allpass}}^*(1/z^*)$ . Therefore, if  $H_{\text{allpass}}(z)$  has a pole at  $z = c$ , then  $H_{\text{allpass}}^*(1/z^*)$  has a zero at  $z = c$ . The latter implies that  $H_{\text{allpass}}(z)$  has a zero at  $z = 1/c^*$ . Therefore, any zero or pole must be accompanied by a matching pole or zero at the *conjugate-reciprocal* location.

### Example 2-9.

A first-order allpass transfer function is given by

$$H_{\text{allpass}}(z) = \frac{z^{-1} - c^*}{1 - cz^{-1}}. \quad (2.39)$$

The pole-zero plot is shown in Figure 2-10. Note that the pole and zero form a conjugate-reciprocal pair. For the value of  $c$  shown in the figure, the impulse response will be complex valued.  $\square$

Observe that  $c$  and  $1/c^*$  have the same angle in the Z plane, but their magnitudes are the reciprocal of one another, as shown in Figure 2-10.

Consider a transfer function of the form

$$H_{\text{allpass}}(z) = z^M \frac{z^{-N} + a_1 z^{-N+1} + \dots + a_N}{1 + a_1^* z^{-1} + \dots + a_N^* z^{-N}}. \quad (2.40)$$

This can be rewritten as

$$H_{\text{allpass}}(z) = \frac{A(z)}{z^{N-M} A^*(1/z^*)}, \quad A(z) = 1 + a_1 z + \dots + a_N z^N. \quad (2.41)$$

Such a transfer function is allpass,

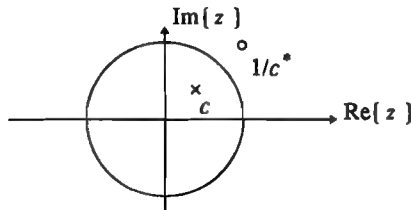


Figure 2-10. The pole-zero plot for a first-order allpass filter.

$$|H_{\text{allpass}}(e^{j\omega T})| = \left| \frac{A(e^{j\omega T})}{e^{j\omega TN} A^*(e^{j\omega T})} \right| = 1. \quad (2.42)$$

Note that if  $N \geq M$ , the term  $z^{N-M}$  in the denominator contributes  $N - M$  poles at  $z = 0$  and  $N - M$  zeros at  $|z| = \infty$ , the conjugate-reciprocal of 0. If  $N < M$ , then this term puts zeros  $z = 0$  and poles at  $|z| = \infty$ . Thus, poles and zeros at zero and infinity also come in conjugate-reciprocal pairs. Since poles and zeros must come in conjugate-reciprocal pairs, any rational allpass transfer function can be written in the form of (2.41).

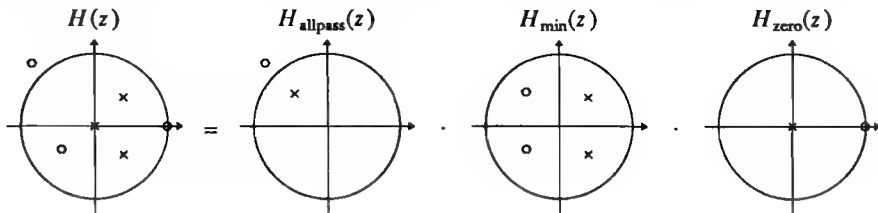
### 2.5.4. Minimum and Maximum-Phase Transfer Functions

A stable and causal transfer function is said to be *strictly minimum-phase* when all its poles and zeros are inside the unit circle. See Problem 2-22 for the intuition that explains the term "minimum phase". A stable and causal transfer function is *strictly maximum-phase* when all poles and zeros are outside the unit circle. Neither strictly minimum-phase nor strictly maximum-phase transfer functions are allowed to have zeros on the unit circle itself. If they have zeros on the unit circle, then we call them *loosely minimum-phase* or *loosely maximum-phase*. The strictly minimum-phase transfer function has the important property that it has a BIBO stable and causal inverse. This is because zeros of the transfer function become poles of the inverse, and for a strictly minimum-phase transfer function, all such poles will end up inside the unit circle. A loosely minimum-phase transfer function does not necessarily have a BIBO stable inverse, but it is still useful because it may display a mild form of instability.

Any causal and stable rational transfer function can be factored as

$$H(z) = H_{\text{allpass}}(z) H_{\text{min}}(z) H_{\text{zero}}(z), \quad (2.43)$$

where  $H_{\text{allpass}}(z)$  is causal, stable, and allpass,  $H_{\text{min}}(z)$  is strictly minimum phase, and  $H_{\text{zero}}(z)$  is FIR with only zeros on the unit circle and corresponding poles at  $z = 0$ . The construction of this factorization, illustrated in Figure 2-11, is straightforward. All zeros of  $H(z)$  on the unit circle are assigned to  $H_{\text{zero}}(z)$ . To ensure that  $H_{\text{zero}}(z)$  is causal, it is given a pole at  $z = 0$  for each zero on the unit circle. Such poles may be



**Figure 2-11.** Factorization of a causal and stable rational transfer function into an allpass transfer function, a strictly minimum-phase transfer function, and a causal transfer function with only zeros on the unit circle. Notice that the number of poles at  $z = 0$  is chosen so that there are no poles at  $z = \infty$ , ensuring that each transfer function is causal.

canceled by zeros at  $z = 0$  in  $H_{\min}(z)$ , if necessary. All remaining poles in  $H(z)$ , which must lie inside the unit circle, are assigned to  $H_{\min}(z)$ . All zeros that lie inside the unit circle are also assigned to  $H_{\min}(z)$ . Each zero outside the unit circle is assigned to  $H_{\text{allpass}}(z)$ . To make sure  $H_{\text{allpass}}(z)$  is allpass, it is assigned a pole at the conjugate-reciprocal location of each such zero. That pole will be inside the unit circle, ensuring that  $H_{\text{allpass}}(z)$  is causal and stable. To cancel the effect of that pole, a zero at the same location is assigned to  $H_{\min}(z)$ . When all is done,  $H_{\min}(z)$  should have an equal number of poles and zeros, all inside the unit circle.

We can develop another useful factorization of a stable (not necessarily causal) rational transfer function by dividing poles and zeros into four classes: those inside, on, and outside the unit circle, plus some zeros at the origin. The factorization is

$$H(z) = B \cdot z^L \cdot H_{\min}(z) \cdot H_{\max}(z) \cdot H_{\text{zero}}(z) \quad (2.44)$$

where  $H_{\min}(z)$  is a strictly minimum-phase transfer function containing all the poles and zeros inside the unit circle, except possibly for some zeros at the origin, while  $H_{\max}(z)$  is a strictly maximum-phase transfer function containing all the poles and zeros outside the unit circle.  $H_{\text{zero}}(z)$  is an FIR transfer function containing all the zeros on the unit circle (stability rules out poles on the unit circle) with corresponding poles at the origin. In addition, we choose constants  $B$  and  $L$  so that  $H_{\min}(z)$  and  $H_{\text{zero}}(z)$  are causal and monic ( $H_{\min}(\infty) = H_{\text{zero}}(\infty) = 1$ ), and  $H_{\max}(z)$  is anti-causal and monic ( $H_{\max}(0) = 1$ ).

With these constraints, the factorization is unique. In particular, the terms can always be written in the form

$$H_{\min}(z) = \frac{\prod_{k=1}^M (1 - c_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})}, \quad |c_k| < 1, \quad |d_k| < 1, \quad (2.45)$$

$$H_{\text{zero}}(z) = \prod_{k=1}^K (1 - e_k z^{-1}), \quad |e_k| = 1, \quad (2.46)$$

$$H_{\max}(z) = \frac{\prod_{k=1}^I (1 - f_k z)}{\prod_{k=1}^J (1 - g_k z)}, \quad |f_k| < 1, \quad |g_k| < 1. \quad (2.47)$$

An example will serve to illustrate how we turn a general rational transfer function into this canonical form.

#### Example 2-10.

Given the rational transfer function

$$H(z) = \frac{(1 - 0.5z^{-1})(1 - z^{-1})}{(1 - 1.25z^{-1})}, \quad (2.48)$$

we can write

$$H(z) = -1.25 z (1 - 0.5z^{-1}) \frac{1}{(1 - 0.8z)} (1 - z^{-1}) . \quad (2.49)$$

We can identify  $B = -1.25$ ,  $L = 1$ , and

$$H_{\min}(z) = (1 - 0.5z^{-1}), H_{\max}(z) = 1/(1 - 0.8z), H_{\text{zero}}(z) = (1 - z^{-1}) . \quad (2.50)$$

□

Given a transfer function in the second or third form of (2.33), the factorization in (2.44) is simple to obtain.

Given a transfer function  $H(z)$ , we define the *reflected* transfer function to be  $H^*(1/z^*)$ . It has impulse response  $h_{-k}^*$  (as is easy to verify). For a rational transfer function in the second form of (2.33), the reflected transfer function can be written

$$H^*(1/z^*) = A \cdot z^{-r} \cdot \frac{\prod_{k=0}^M (1 - c_k^* z)}{\prod_{k=0}^N (1 - d_k^* z)} . \quad (2.51)$$

The zeros  $c_k$  of  $H(z)$  become zeros at the conjugate-reciprocal locations  $1/c_k^*$  in  $H^*(1/z^*)$ . The poles are similarly reflected through the unit circle. If  $H(z)$  is minimum-phase and monic, then  $H^*(1/z^*)$  is maximum-phase and monic, and *vice versa*. If  $H(z)$  is stable and causal (all poles are inside the unit circle) then  $H^*(1/z^*)$  is stable and anti-causal (all poles are outside the unit circle). Zeros of  $H(z)$  on the unit circle have corresponding zeros of  $H^*(1/z^*)$  at identical locations on the unit circle.

To see the relationship between the frequency response of  $H(z)$  and  $H^*(1/z^*)$ , just evaluate them at  $z = e^{j\omega T}$ , getting  $H(e^{j\omega T})$  and  $H^*(e^{j\omega T})$ . Hence the frequency response of a reflected transfer function is simply the complex-conjugate of the original frequency response. That is, any transfer function and its reflected transfer function have the same magnitude frequency response, and their phase responses are the negative of one another.

#### Example 2-11.

Since  $H(z)$  and  $H^*(1/z^*)$  have the same magnitude response on the unit circle, the system  $H_{\text{allpass}}(z) = H(z)/H^*(1/z^*)$  is an allpass system. □

Using the factorization in (2.44) and Example 2-11 is an alternative route to the factorization of (2.43) (see Problem 2-24).

### 2.5.5. Non-Negative Real Transfer Functions

In the study of random processes (Chapter 3), Z transforms that are real valued and non-negative on the unit circle arise frequently. In this subsection, we study the properties of such Z transforms.

Suppose  $H(z)$  is a rational transfer function, and  $S(z)$  is



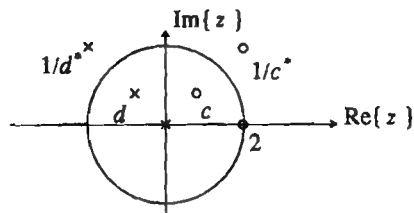


Figure 2-12. An example of a pole zero plot for an  $S(z)$  that is real valued on the unit circle.

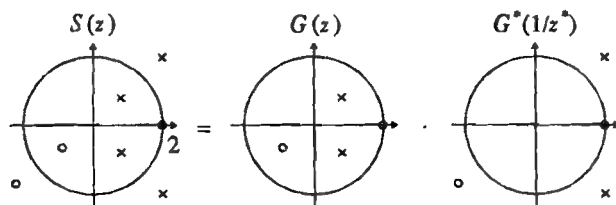


Figure 2-13. Spectral factorization of a transfer function  $S(z)$ , which is non-negative real on the unit circle. The zero at  $z = 1$  has multiplicity two (in general its multiplicity could be any even integer). These two zeros on the unit circle are split between  $G(z)$  and  $G^*(1/z^*)$ .

$$S(z) = H(z)H^*(1/z^*) . \quad (2.52)$$

Then  $S(z)$  is always non-negative and real valued on the unit circle,

$$S(z) = H(z)H^*(1/z^*) \Big|_{z=e^{j\omega T}} = |H(e^{j\omega T})|^2 . \quad (2.53)$$

For  $S(z)$  in (2.52), for every zero at  $z_0$ , there is a zero at  $1/z_0^*$ . This follows trivially from the observation that

$$S(z) = S^*(1/z^*) . \quad (2.54)$$

Similarly, the poles must come in conjugate-reciprocal pairs.

#### Example 2-12.

An example pole-zero plot for a transfer function of form (2.52) is shown in Figure 2-12. The zero at  $z = c$  has a matching zero at  $z = 1/c^*$ . The pole at  $z = d$  also has a matching pole. The double zero at  $z = 1$  illustrates another implication of (2.52); any zero on the unit circle must be double. The pole at  $z = 0$  has a matching pole at  $|z| = \infty$ . Although the latter pole is not explicitly shown, it is implied because only three poles are shown, compared to four zeros. Note that the impulse response  $s_k$  is not real for this example, but nonetheless the frequency response  $S(e^{j\omega T})$  is.  $\square$

The conjugate-reciprocal symmetry of (2.52) is not the same as that found for allpass filters. It has conjugate-symmetric pole pairs and zero pairs, rather than pole-zero pairs as in the allpass filter. While allpass filters can be causal, only a trivial non-negative real transfer function in the form of (2.52) can be causal.

It is shown in Appendix 2-B that all stable rational transfer functions  $S(z)$  that are non-negative real valued on the unit circle can be written in the form of (2.52); more strongly,  $S(z)$  can be written as

$$S(z) = A^2 G(z) G^*(1/z^*) \quad (2.55)$$

where  $A$  is some real-valued constant, and  $G(z)$  is a loosely minimum-phase rational transfer function,

$$G(z) = \frac{\prod_{k=1}^M (1 - c_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})}, \quad |c_k| \leq 1, \quad |d_k| < 1. \quad (2.56)$$

$S(z)$  can thus be written as the product of a loosely minimum-phase monic transfer function and its reflected transfer function. The constant  $A$  is chosen so that  $G(z)$  is monic, and turns out to be quite important.

Equation (2.55) is the *monic minimum-phase spectral factorization* of  $S(z)$ . It is obtained from  $S(z)$  by accumulating within  $G(z)$  all the poles and zeros of  $S(z)$  within the unit circle, plus one of each double-zero pair of  $S(z)$  on the unit circle. This factorization is illustrated in Figure 2-13, where we have, from left to right, the original  $S(z)$ , the loosely minimum-phase term (poles and zeros inside or on the unit circle), and its reflected transfer function.

A remarkable fact (derived in Appendix 2-B) is that  $A^2$  is equal to the *geometric mean* of  $S(e^{j\omega T})$ ,

$$A^2 = \exp \left\{ \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \ln [S(e^{j\omega T})] d\omega \right\}, \quad (2.57)$$

where  $\ln(\cdot)$  is the natural logarithm, equal to  $\log_e(\cdot)$ .

## 2.6. SIGNAL SPACE REPRESENTATIONS

It is possible to abstractly represent the signals in a digital communication system as vectors in a *linear space* or *vector space*, much like the familiar three-dimensional vectors in our physical world. This representation does not allow us to solve any problems that we cannot solve by other methods, but it gives valuable intuition. The linear space used in our context is often called *signal space*, since vectors in the space represent signals.

### 2.6.1. Definition of Signal Space

Formally, a *linear space* or *vector space* is a set of vectors together with two operators, addition of vectors and multiplication by a scalar.

**Example 2-13.**

Ordinary *Euclidean space* is the most familiar example of a linear space. In Euclidean space, a vector is specified by its coordinates,  $n$  coordinates in an  $n$ -dimensional space,

$$\mathbf{X} \leftrightarrow (x_1, x_2, \dots, x_n), \quad (2.58)$$

where  $\mathbf{X}$  is the vector and  $x_1, \dots, x_n$  are the  $n$  components of that vector. The notation " $\leftrightarrow$ " means that  $\mathbf{X}$  is the vector which corresponds to components  $x_1, \dots, x_n$ . There are rules for adding two vectors (sum the individual components) and multiplying a vector by a scalar (multiply each of the components by that scalar).  $\square$

**Example 2-14.**

A space of some importance in this book is the *Euclidean space of complex-valued vectors*. Vectors in this space are identical to (2.58) except that the components  $x_k$  of the vector are complex-valued. Ordinary Euclidean space is of course a special case of this, where the imaginary parts of the vectors are zero.  $\square$

The addition rule produces a new vector  $\mathbf{X} + \mathbf{Y}$  that must be in the linear space. Addition must obey familiar rules of arithmetic, such as the commutative and associative laws,

$$\mathbf{X} + \mathbf{Y} = \mathbf{Y} + \mathbf{X}, \quad \mathbf{X} + (\mathbf{Y} + \mathbf{Z}) = (\mathbf{X} + \mathbf{Y}) + \mathbf{Z}. \quad (2.59)$$

The direct sum of two vectors has the interpretation illustrated in Figure 2-14a for the two-dimensional Euclidean space. A linear space must include a zero vector  $\mathbf{0}$ , and every vector must have an *additive inverse*, denoted  $-\mathbf{X}$ , such that

$$\mathbf{0} + \mathbf{X} = \mathbf{X}, \quad \mathbf{X} + (-\mathbf{X}) = \mathbf{0}. \quad (2.60)$$

Multiplication by a scalar  $\alpha$  produces a new vector  $\alpha \cdot \mathbf{X}$  that must be in the vector space. Multiplications must obey the associative law,

$$\alpha \cdot (\beta \cdot \mathbf{X}) = (\alpha\beta) \cdot \mathbf{X} \quad (2.61)$$

and also follow the rules

$$1 \cdot \mathbf{X} = \mathbf{X}, \quad 0 \cdot \mathbf{X} = \mathbf{0}. \quad (2.62)$$

The geometric interpretation of multiplying a vector by a scalar is shown in Figure 2-14b. Finally, addition and multiplication must obey the distributive laws,

$$\alpha \cdot (\mathbf{X} + \mathbf{Y}) = \alpha \cdot \mathbf{X} + \alpha \cdot \mathbf{Y}, \quad (\alpha + \beta) \cdot \mathbf{X} = \alpha \cdot \mathbf{X} + \beta \cdot \mathbf{X}. \quad (2.63)$$

*Real* linear spaces have real-valued scalars as components of vectors, while *complex* linear spaces have complex-valued components. We will encounter both types.

Euclidean space as defined earlier meets all of these requirements, and is therefore a linear space. There are two other examples of linear spaces of particular importance in communication theory: the space of discrete-time signals (which is a generalization of Euclidean space to infinite dimensions), and the space of continuous-time signals. Since these linear spaces model the two basic types of signals we encounter in digital communication systems, we call them *signal spaces*.

**Example 2-15.**

Given a complex-valued discrete-time signal  $\{y_k\}$ , define a vector

$$\mathbf{Y} \leftrightarrow (\dots y_{-1}, y_0, y_1, \dots) . \quad (2.64)$$

The set of all such vectors is similar to Euclidean space as defined in (2.58), the difference being that the number of components is infinite rather than finite. An additional assumption often made is that

$$\sum_k |y_k|^2 < \infty , \quad (2.65)$$

or, in words, that the total energy in the discrete-time signal is finite. This assumption is necessary for mathematical reasons that will become evident shortly. Scalar multiplication and vector addition are the same as for Euclidean spaces.  $\square$

**Example 2-16.**

Define a vector  $\mathbf{Y}$  to correspond to a continuous-time signal  $y(t)$ ,

$$\mathbf{Y} \leftrightarrow y(t), -\infty < t < \infty , \quad (2.66)$$

where as in (2.65), there is an assumption of finite energy,

$$\int_{-\infty}^{\infty} |y(t)|^2 dt < \infty . \quad (2.67)$$

We can think of this space as a strange Euclidean space with a continuum of coordinates. The definition of multiplication of a signal vector by a scalar and the summation of two signal vectors are the obvious,

$$\alpha \cdot \mathbf{Y} \leftrightarrow \alpha y(t) , \quad \mathbf{X} + \mathbf{Y} \leftrightarrow x(t) + y(t) , \quad (2.68)$$

and the definition of a zero vector is the zero-valued signal.  $\square$

**Exercise 2-13.**

Verify that the linear spaces given by Example 2-15 and Example 2-16 satisfy the properties of (2.59) through (2.63).  $\square$

The following example relates these somewhat abstract concepts to a simple digital communication system.

**Example 2-17.**

In a digital communication system, suppose that we want to transmit and receive a single data symbol  $A$ , where  $A$  assumes a small number of values, for example two values in a binary system. For maximum generality consider  $A$  to be complex-valued, although the physical meaning of this will not become evident until Chapter 6. In a form of modulation called *pulse amplitude modulation (PAM)* (covered in more detail in Chapter 6), the amplitude of a transmitted pulse  $h(t)$  is multiplied by the transmitted data symbol  $A$ . The transmitted signal is therefore of the form

$$x(t) = A h(t) . \quad (2.69)$$

In accordance with our linear space notation, we can associate the transmitted pulse  $h(t)$  with a vector in signal space

$$\mathbf{H} \leftrightarrow h(t) \quad (2.70)$$

in which case the transmitted signal corresponds to the vector

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{H} . \quad (2.71)$$

□

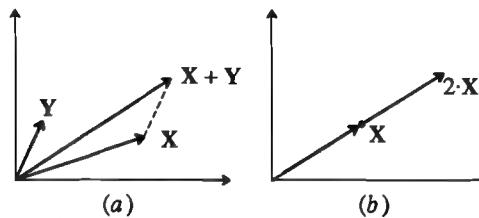
### 2.6.2. Geometric Structure of Signal Space

The definition of a linear space does not capture the most important properties of Euclidean space; namely, its *geometric* structure. This structure includes such concepts as the length of a vector in the space, and the angle between two vectors. All these properties of Euclidean space can be deduced from the definition of *inner product* of two vectors. The inner product is defined to be

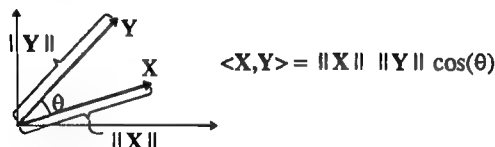
$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^n x_i y_i^* \quad (2.72)$$

for an  $n$ -dimensional Euclidean space, where  $y_i^*$  is the complex conjugate of  $y_i$ . It has the interpretation illustrated in Figure 2-15; namely, the inner product of two vectors is equal to the product of the length of the first vector, the length of the second vector, and the cosine of the angle between the vectors.

In Figure 2-15,  $\|\mathbf{X}\|$  denotes the length of a vector, which has not been defined. However, once the definition of inner product (2.72) has been given, we can *deduce* a reasonable definition for the length of a vector, since the inner product of a vector with itself,  $\langle \mathbf{X}, \mathbf{X} \rangle$ , is the square of the length of the vector (the angle  $\theta$  is zero). A



**Figure 2-14.** Elementary operations in a two-dimensional linear space. a. Sum of two vectors. b. Multiplication of a vector by a scalar.



**Figure 2-15.** Geometrical interpretation of inner product.

special notation is used for  $\langle \mathbf{X}, \mathbf{X} \rangle$ ,

$$\langle \mathbf{X}, \mathbf{X} \rangle = \|\mathbf{X}\|^2 = \sum_{i=1}^n |x_i|^2 \quad (2.73)$$

where  $\|\mathbf{X}\|$  is called the *norm* of the vector  $\mathbf{X}$  and geometrically is the length of the vector. This notation is used in Figure 2-15.

Note that  $\|\mathbf{Y}\|\cos(\theta)$  is the length of the component of  $\mathbf{Y}$  in the direction of  $\mathbf{X}$ . Hence we get a particularly useful interpretation of the inner product:  $\langle \mathbf{X}, \mathbf{Y} \rangle / \|\mathbf{X}\|$  is the length of the component of  $\mathbf{Y}$  in the direction of  $\mathbf{X}$ , and  $\langle \mathbf{X}, \mathbf{Y} \rangle / \|\mathbf{Y}\|$  is the length of the component of  $\mathbf{X}$  in the direction of  $\mathbf{Y}$ .

Two vectors  $\mathbf{X}$ ,  $\mathbf{Y}$  are said to be *orthogonal* if

$$\langle \mathbf{X}, \mathbf{Y} \rangle = 0. \quad (2.74)$$

This means that  $\mathbf{X}$  has no component in the direction of  $\mathbf{Y}$  and vice versa; they are at right angles! This concept is crucial to the understanding of optimum receiver design in digital communication systems.

The inner product as applied to Euclidean space can be generalized to the other linear spaces of interest. The important consequence is that the geometric concepts familiar in Euclidean space can be applied to these spaces as well. Let  $\mathbf{X}$  and  $\mathbf{Y}$  be vectors of a linear space on which an inner product  $\langle \mathbf{X}, \mathbf{Y} \rangle$  is defined. The inner product is a scalar (complex-valued number), and must obey the rules

$$\langle \mathbf{X} + \mathbf{Y}, \mathbf{Z} \rangle = \langle \mathbf{X}, \mathbf{Z} \rangle + \langle \mathbf{Y}, \mathbf{Z} \rangle \quad (2.75)$$

$$\langle \alpha \mathbf{X}, \mathbf{Y} \rangle = \alpha \langle \mathbf{X}, \mathbf{Y} \rangle, \quad \langle \mathbf{X}, \mathbf{Y} \rangle = \langle \mathbf{Y}, \mathbf{X} \rangle^* \quad (2.76)$$

$$\langle \mathbf{X}, \mathbf{X} \rangle > 0, \text{ for } \mathbf{X} \neq 0. \quad (2.77)$$

These rules are all obeyed by the familiar Euclidean space inner product of (2.72), as can be easily verified. For the other linear spaces of interest, analogous definitions of the inner product satisfying the rules can be made. In particular, define the inner product and norm (deduced from (2.73)) of two discrete-time signals as

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{k=-\infty}^{\infty} x_k y_k^*, \quad \|\mathbf{X}\|^2 = \sum_{k=-\infty}^{\infty} |x_k|^2 \quad (2.78)$$

and of two continuous-time signals as

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \int_{-\infty}^{\infty} x(t) y^*(t) dt, \quad \|\mathbf{X}\|^2 = \int_{-\infty}^{\infty} |x(t)|^2 dt. \quad (2.79)$$

These inner products can be given the same interpretation as in Euclidean space; namely, as the product of the length of two vectors times the cosine of the angle between them. Thus, the inner product serves to define the "angle" between two vectors.

#### Exercise 2-14.

Verify that the definitions of inner product of (2.78) and (2.79) satisfy the properties of

(2.75) through

(2.77).  $\square$

Note that conditions (2.65) and (2.67) that were imposed correspond to the assumption that a vector has finite norm or length. This explains the need for these initial assumptions in the definition of the linear spaces.

### Example 2-18.

Consider again the simple digital communication system of Example 2-17 in which a single transmitted pulse  $h(t)$  is multiplied by a data symbol  $A$ . Suppose that this transmitted waveform is corrupted by noise before arriving at the receiver, and we decide to implement in the receiver a filter which rejects as much of this noise as possible. In particular, as shown in Figure 2-16, we implement a filter with impulse response  $h^*(-t)$ , the conjugate mirror image of the transmitted pulse, with corresponding frequency response  $H^*(j\omega)$ . As we will see in Chapters 6-8, this is not as arbitrary as it may seem, since this particular filter is an optimum filter to reject noise in a special sense, and is given the special name *matched filter*. The output of this filter is sampled at time  $t = 0$ , resulting in the value

$$y(0) = \int_{-\infty}^{\infty} x(t)h^*(t) dt = \langle \mathbf{X}, \mathbf{H} \rangle. \quad (2.80)$$

The inner product operation is interpreted geometrically as the component of the received signal  $\mathbf{X}$  in the direction of the transmitted signal vector  $\mathbf{H}$  (multiplied by the unimportant constant  $\|\mathbf{H}\|$ ). Intuitively this seems to be a reasonable approach, since components in directions other than that of the transmitted signal may be irrelevant. Thus the optimality of the matched filter is not surprising from a geometric point of view.  $\square$

The geometric properties are so important that the special name *inner product space* is given to a linear space on which an inner product is defined. Thus, both Example 2-15 and Example 2-16 defined earlier are inner product spaces. If the inner product space has the additional property of *completeness*, then it is defined to be a *Hilbert space*. Intuitively the notion of completeness means that there are no "missing" vectors that are arbitrarily close to vectors in the space but are not themselves in the space. Since the spaces used in this book are all complete and hence formally Hilbert spaces, we will not dwell on this property further. In the sequel, all linear spaces considered will be Hilbert spaces.

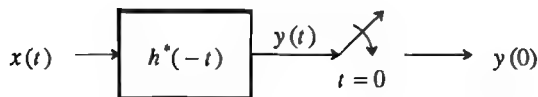


Figure 2-16. A matched filter.

### 2.6.3. Subspaces of Signal Space

A *subspace* of a linear space is a subset of the linear space that is itself a linear space. Roughly speaking this means that the sum of any two vectors in the subspace must also be in the subspace, and the product of any vector in the subspace by any scalar must also be in the subspace.

#### Example 2-19.

An example of a subspace in three-dimensional Euclidean space is either a line or a plane in the space, where in either case the vector  $\mathbf{0}$  must be in the subspace.  $\square$

#### Example 2-20.

A more general subspace is the set of vectors obtained by forming all possible weighted linear combinations of  $n$  vectors  $\mathbf{X}_1, \dots, \mathbf{X}_n$ . The subspace so formed is said to be *spanned* by the set of  $n$  vectors. This is illustrated in Figure 2-17 for three-dimensional Euclidean space. In Figure 2-17a, the subspace spanned by  $\mathbf{X}$  is the dashed line, which is infinite in length and co-linear with the vector  $\mathbf{X}$ . Any vector on this line can be obtained by multiplying  $\mathbf{X}$  by the appropriate scalar. In Figure 2-17b, the subspace spanned by  $\mathbf{X}$  and  $\mathbf{Y}$  is the plane of infinite extent (depicted by the dashed lines) that is determined by the two vectors. Any vector in this plane can be formed as a linear combination of the two vectors multiplied by appropriate scalars.  $\square$

The *projection theorem* is an important result that can often be used to derive optimum filters and estimators. What follows is a statement of the projection theorem, which is proven in [1]:

**(Projection Theorem)** Given a subspace  $M$  of a Hilbert space  $H$  and a vector  $\mathbf{X}$  in  $H$  there is a unique vector  $\mathbf{P}_M(\mathbf{X})$  in  $M$  called the *projection of  $\mathbf{X}$  on  $M$*  which has the property that

$$\langle \mathbf{X} - \mathbf{P}_M(\mathbf{X}), \mathbf{Y} \rangle = 0 \quad (2.81)$$

for every vector  $\mathbf{Y}$  in  $M$ . The notation  $\mathbf{P}_M$  denotes a *projection operator* that maps one vector  $\mathbf{X}$  into another vector  $\mathbf{P}_M(\mathbf{X})$ .

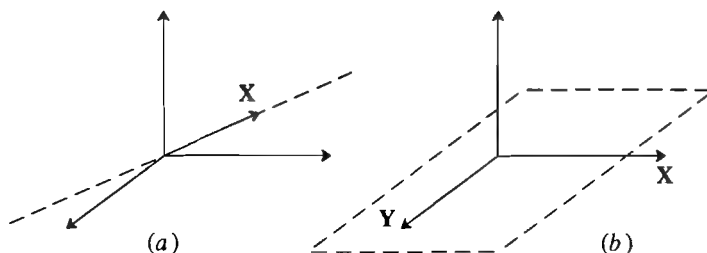
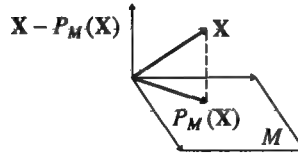


Figure 2-17. Subspaces in three-dimensional Euclidean space.





**Figure 2-18.** Illustration of projection for three-dimensional Euclidean space.

**Example 2-21.**

A projection is illustrated in Figure 2-18 for three-dimensional Euclidean space, where the subspace  $M$  is the plane formed by the  $x$ -axis and  $y$ -axis and  $X$  is an arbitrary vector. The projection is the result of dropping a perpendicular line from  $X$  down to the plane (this is the dashed line in Figure 2-18). The resulting vector  $(X - P_M(X))$  is the vector shown parallel to the dashed line. It is orthogonal to the plane  $M$ , and hence to every vector in  $M$ .  $\square$

A consequence of the projection theorem is that the projection  $P_M(X)$  is the unique vector in  $M$  that is closest to  $X$ .

**Exercise 2-15.**

Show that

$$\|X - P_M(X)\| < \|X - Y\| \quad (2.82)$$

for every  $Y \neq P_M(X)$  in  $M$ .  $\square$

This is illustrated geometrically in Figure 2-18, where the subspace  $M$  is a plane. The vector closest to  $X$  is evidently the projection as shown, and any other vector in  $M$  is farther from  $X$ .

## 2.6.4. Schwarz Inequality

A fundamental inequality that often gives bounds on the performance of a digital communication system is the *Schwarz inequality*.

**Exercise 2-16.**

Show that for two vectors  $X$  and  $Y$  in an inner product space,

$$|\langle X, Y \rangle| \leq \|X\| \cdot \|Y\| \quad (2.83)$$

with equality if and only if  $X = K \cdot Y$  for some scalar  $K$ .  $\square$

## 2.7. FURTHER READING

Many textbooks cover the topics of this chapter in a more introductory and complete fashion than we do here. McGillem and Cooper [2], Oppenheim and Willsky [3], and Ziemer, Tranter, and Fannin [4] are useful for techniques applicable to both continuous and discrete-time systems. For discrete-time techniques only, the texts by Oppenheim and Schaffer [5] and Jackson [6] are recommended. For continuous-time systems, with some discussion of discrete-time systems, we recommend Schwarz and Friedland [7]. To explore the Fourier transform in more mathematical depth, we recommend Papoulis [8] and Bracewell [9].

### APPENDIX 2-A SUMMARY OF FOURIER TRANSFORM PROPERTIES

The properties of both discrete and continuous-time Fourier transforms are summarized in this appendix. We define the even part  $f_e(x)$  of a function  $f(x)$  to be

$$f_e(x) = [f(x) + f^*(-x)]/2, \quad (2.84)$$

and the odd part  $f_o(x)$  to be

$$f_o(x) = [f(x) - f^*(-x)]/2, \quad (2.85)$$

so for example,

$$X_e(e^{j\omega T}) = [X(e^{j\omega T}) + X^*(e^{-j\omega T})]/2. \quad (2.86)$$

We define the rectangular function as follows,

$$\text{rect}(x, X) = \begin{cases} 1; & |x| \leq X \\ 0; & |x| > X \end{cases}, \quad (2.87)$$

and the unit step function as

$$u(x) = \begin{cases} 1; & |x| \geq 0 \\ 0; & |x| < 0 \end{cases}. \quad (2.88)$$

## FOURIER TRANSFORM SYMMETRIES

Continuous time	Discrete time
$x(t) \leftrightarrow X(j\omega)$	$x_k \leftrightarrow X(e^{j\omega T})$
$x(-t) \leftrightarrow X(-j\omega)$	$x_{-k} \leftrightarrow X(e^{-j\omega T})$
$x^*(t) \leftrightarrow X^*(-j\omega)$	$x_k^* \leftrightarrow X^*(e^{-j\omega T})$
$x^*(-t) \leftrightarrow X^*(j\omega)$	$x_{-k}^* \leftrightarrow X^*(e^{j\omega T})$
$\text{Re}\{x(t)\} \leftrightarrow X_e(j\omega)$	$\text{Re}\{x_k\} \leftrightarrow X_e(e^{j\omega T})$
$j\text{Im}\{x(t)\} \leftrightarrow X_o(j\omega)$	$j\text{Im}\{x_k\} \leftrightarrow X_o(e^{j\omega T})$
$x_e(t) \leftrightarrow \text{Re}\{X(j\omega)\}$	$x_{e,k} \leftrightarrow \text{Re}\{X(e^{j\omega T})\}$
$x_o(t) \leftrightarrow j\text{Im}\{X(j\omega)\}$	$x_{o,k} \leftrightarrow j\text{Im}\{X(e^{j\omega T})\}$

## FOURIER TRANSFORMS PROPERTIES

Continuous time	Discrete time
$ax(t) + by(t) \leftrightarrow aX(j\omega) + bY(j\omega)$	$ax_k + by_k \leftrightarrow aX(e^{j\omega T}) + bY(e^{j\omega T})$
$x(t) * y(t) \leftrightarrow X(j\omega)Y(j\omega)$	$x_k * y_k \leftrightarrow X(e^{j\omega T})Y(e^{j\omega T})$
$x(t)y(t) \leftrightarrow \frac{1}{2\pi} X(j\omega) * Y(j\omega)$	$x_k y_k \leftrightarrow \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} X(e^{j\Omega T}) Y(e^{j(\omega-\Omega)T}) d\Omega$
$x(at) \leftrightarrow \frac{1}{ a } X(j\frac{\omega}{a})$	$x_{k-K} \leftrightarrow X(e^{j\omega T}) e^{-j\omega K T}$
$x(t - \tau) \leftrightarrow X(j\omega) e^{-j\omega\tau}$	$e^{j\omega_0 k T} x_k \leftrightarrow X(e^{j(\omega - \omega_0)T})$
$e^{j\omega_0 t} x(t) \leftrightarrow X(j\omega - j\omega_0)$	$\cos(\omega_0 k T) x_k \leftrightarrow \frac{1}{2} (X(e^{j(\omega - \omega_0)T}) + X(e^{j(\omega + \omega_0)T}))$
$\cos(\omega_0 t) x(t) \leftrightarrow \frac{1}{2} (X(j\omega - j\omega_0) + X(j\omega + j\omega_0))$	
$\frac{d^m x(t)}{dt^m} \leftrightarrow (j\omega)^m X(j\omega)$	
$(-jt)^m x(t) \leftrightarrow \frac{d^m X(j\omega)}{d\omega^m}$	
$\int_{-\infty}^t x(\tau) d\tau \leftrightarrow \frac{1}{j\omega} X(j\omega) + \pi\delta(\omega) \int_{-\infty}^{\infty} x(\tau) d\tau$	
$X(j\omega) \leftrightarrow 2\pi x(-\omega)$	

FOURIER TRANSFORM PAIRS<sup>1</sup>

$e^{j\omega t} \leftrightarrow 2\pi\delta(\omega - \omega_0)$	$e^{j\omega kT} \leftrightarrow \frac{2\pi}{T}\delta(\omega - \omega_0)$
$\delta(t - T) \leftrightarrow e^{-j\omega T}$	$\delta_k - \pi \leftrightarrow e^{-j\omega kT}$
$\cos(\omega_0 t) \leftrightarrow \pi [\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$	$\cos(\omega_0 kT) \leftrightarrow \frac{\pi}{T}(\delta(\omega - \omega_0) + \delta(\omega + \omega_0))$
$\sin(\omega_0 t) \leftrightarrow \frac{1}{j}\pi [\delta(\omega - \omega_0) - \delta(\omega + \omega_0)]$	$\sin(\omega_0 kT) \leftrightarrow \frac{\pi}{jT}(\delta(\omega - \omega_0) - \delta(\omega + \omega_0))$
$\frac{\sin(Wt)}{Wt} \leftrightarrow \frac{\pi}{W}\text{rect}(\omega, W)$	$\frac{\sin(WkT)}{WkT} \leftrightarrow \frac{\pi}{WT}\text{rect}(\omega, W)$
$e^{-\alpha} u(t) \leftrightarrow \frac{1}{j\omega + \alpha}; \text{Re}\{\alpha\} > 0$	$r^{-k} u_k \leftrightarrow \frac{1}{1 - r^{-1}e^{-j\omega T}}$
$u(t) \leftrightarrow \pi\delta(\omega) + \frac{1}{j\omega}$	
$\sum_{k=-\infty}^{\infty} \delta(t - kT) \leftrightarrow \frac{2\pi}{T} \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi}{T}m)$	$\sum_{m=-\infty}^{\infty} \delta_k - mN \leftrightarrow \frac{2\pi}{NT} \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi}{NT}m)$
$\text{rect}(t, T) \leftrightarrow 2T \frac{\sin(\omega T)}{\omega T}$	
$\frac{1}{jt} \leftrightarrow -\pi \text{sgn}(\omega)$	

## APPENDIX 2-B

### SPECTRAL FACTORIZATION

In this appendix we will derive the spectral factorization (2.55) of a rational transfer function that is non-negative real on the unit circle, and also derive the geometric mean representation of  $A^2$ .

#### Exercise 2-17.

The purpose of this exercise is to show that any transfer function  $S(z)$  that is real valued (not necessarily non-negative) on the unit circle, must have conjugate-reciprocal pole pairs and zero pairs.

- Show that if  $S(e^{j\omega T})$  is real valued for all  $\omega$ , then the inverse Fourier transform  $s_k$  is conjugate symmetric,  $s_k = s_{-k}^*$ .
- Show that the symmetry relationship in (a) implies (2.54). Hence, (2.54) is valid for any  $S(z)$  that is real valued on the unit circle.  $\square$

We can now study how the general factorization of (2.44) is modified for the non-negative real transfer function. Equation (2.44) tells us that for any stable  $S(z)$  there exist monic strictly minimum-phase and strictly maximum-phase transfer functions

<sup>1</sup> The discrete-time Fourier transform expressions are valid in the range  $-\pi/T \leq \omega \leq \pi/T$ . To extend this range, the given expression should be repeated periodically.

such that

$$S(z) = B \cdot z^L H_{\min}(z) H_{\max}(z) H_{\text{zero}}(z), \quad (2.89)$$

where  $H_{\min}(z)$  includes all zeros and poles inside the unit circle,  $H_{\max}(z)$  includes all zeros and poles outside the unit circle, and  $H_{\text{zero}}(z)$  includes all zeros on the unit circle. Exercise 2-17 implies that  $H_{\min}(z) = H_{\max}^*(1/z^*)$ , since poles and zeros come in conjugate-reciprocal pairs. Thus, the minimum-phase and maximum-phase parts are each reflected transfer functions of the other. Since they are reflected, we know that they are the complex conjugate of one another on the unit circle, and hence the contribution of  $H_{\min}(z) H_{\max}(z)$  is real and non-negative on the unit circle.

Unfortunately, Exercise 2-17 does not tell us anything new about  $H_{\text{zero}}(z)$ , since for  $|z| = 1$ , it is automatically true that  $z = 1/z^*$ . We thus have to investigate further the nature of  $B \cdot z^L H_{\text{zero}}(z)$ . In particular, we are interested in whatever restrictions there are on its zeros in order for it to be real valued, or non-negative real valued.

#### Exercise 2-18.

Establish the following necessary and sufficient conditions on  $B \cdot z^L H_{\text{zero}}(z)$  to be real valued on the unit circle: if its zeros are at  $z_i = e^{j\theta_i}$ ,  $1 \leq i \leq K$ , then we must have that  $K = 2L$  (the number of zeros on the unit circle must be even), and the constant coefficient  $B$  must be of the form

$$B = C \exp\left\{-j \sum_{i=1}^{2L} \theta_i/2\right\} \quad (2.90)$$

for any real-valued constant  $C$ .  $\square$

The role of the  $z^L$  term is to force  $z^L H_{\text{zero}}(z)$  to have the same number of terms in positive and negative powers of  $z$  (recall that  $H_{\text{zero}}(z)$  is by assumption causal and monic, and hence only has non-positive powers of  $z$ ). Exercise 2-18 says that any transfer function with zeros on the unit circle is real valued, as long as the number of zeros is even, the constant coefficient has the proper phase, and it is multiplied by the proper power of  $z$ . Exercise 2-18 still doesn't answer the question of when  $B \cdot z^L H_{\text{zero}}(z)$  is non-negative real valued.

#### Exercise 2-19.

Show that when the conditions of Exercise 2-18 are satisfied, the resulting transfer function is non-negative real valued on the unit circle if and only if it has  $L$  double zeros and the constant  $C$  is of the form  $C = (-1)^L A^2$  where  $A$  is real valued.  $\square$

We can now assert that if  $S(z)$  is non-negative real,

$$\begin{aligned} B \cdot z^L H_{\text{zero}}(z) &= (-1)^L A^2 \prod_{i=1}^L e^{-j\theta_i} z^L \prod_{i=1}^L (1 - e^{j\theta_i} z^{-1})^2 \\ &= A^2 \prod_{i=1}^L (1 - e^{-j\theta_i} z) (1 - e^{j\theta_i} z^{-1}), \end{aligned} \quad (2.91)$$

which is of the form of the product of a transfer function times its reflected transfer

function. Hence, combining (2.91) with (2.89), we obtain the spectral factorization (2.55).

To study the multiplicative constant  $A^2$ , replace  $z$  by  $(e^{j\omega T})$  in (2.56) to get

$$S(e^{j\omega T}) = A^2 \frac{\prod_{k=1}^M |1 - c_k e^{-j\omega T}|^2}{\prod_{k=1}^N |1 - d_k e^{-j\omega T}|^2}, \quad |c_k| \leq 1, \quad |d_k| < 1. \quad (2.92)$$

Taking the logarithm of both sides (the base does not matter), and then integrating over the full Nyquist bandwidth,

$$\begin{aligned} \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \log S(e^{j\omega T}) d\omega &= \log A^2 + \sum_{k=1}^M \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \log |1 - c_k e^{-j\omega T}|^2 d\omega \\ &\quad - \sum_{k=1}^N \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \log |1 - d_k e^{-j\omega T}|^2 d\omega. \end{aligned} \quad (2.93)$$

Fortunately, the last two terms are zero. To see this, write  $c_k$  or  $d_k$  in polar form as  $a e^{j\theta}$ , where  $0 < |a| \leq 1$ . Then we wish to evaluate the integral

$$\frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \log |1 - a e^{j\theta} e^{-j\omega T}|^2 d\omega. \quad (2.94)$$

After some manipulation (see Problem 2-30), this becomes

$$\frac{1}{\pi} \int_0^\pi \log (1 + a^2 - 2a \cos \omega) d\omega = 0. \quad (2.95)$$

Note that the angle  $\theta$  of the pole or zero does not affect the integral. Integral (2.95) can be found in standard integral tables, which show that it evaluates to zero. Thus, we have established (2.57). While (2.57) was derived only for rational spectra, both the spectral factorization (2.55) and the geometric mean formula (2.57) apply to general (non-rational) spectra as well.

## PROBLEMS

- 2-1. A system with a complex-valued input and output can be described in terms of systems with real-valued inputs and outputs, as shown in Figure 2-2. Show that if the impulse response of the system is real-valued, then there is no crosstalk (or cross-coupling) between the real and imaginary parts, whereas if the impulse response is complex-valued then there is crosstalk.
- 2-2.
- Show that  $e^{j\omega t}$  is an *eigenfunction* of a continuous-time LTI system with impulse response  $h(t)$ , meaning that the response to this input is the same complex exponential multiplied by a complex constant called the *eigenvalue*.
  - Repeat for a discrete-time LTI system with impulse response  $h_k$ .

- (c) Show that for a fixed  $\omega$  the eigenvalue in (b) is the Fourier transform  $H(e^{j\omega T})$  of the discrete-time impulse response  $h_k$ . Specifically, show that when the input is  $e^{j\omega kT}$ , the output can be written

$$y_k = H(e^{j\omega T})e^{j\omega kT}. \quad (2.96)$$

Hence that magnitude response  $|H(e^{j\omega T})|$  gives the gain of the system at each frequency, and the phase response  $\arg(H(e^{j\omega T}))$  gives the phase change.

- 2-3. Consider the mixed discrete and continuous-time system in Figure 2-19.

- (a) Find the Fourier transform of  $y(t)$ .  
 (b) Is the system linear? Justify.  
 (c) Find conditions on  $G(j\omega)$ ,  $H(e^{j\omega T})$ , and/or  $F(j\omega)$  such that the system is time invariant.

- 2-4. Derive the following Parseval's relationships for the energy of a signal:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(j\omega)|^2 d\omega, \quad \sum_{m=-\infty}^{\infty} |x_m|^2 = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} |X(e^{j\omega T})|^2 d\omega. \quad (2.97)$$

- 2-5. Given that a discrete-time signal  $x_k$  is obtained from a continuous-time signal  $x(t)$  by sampling, can you relate the energy of the discrete-time signal to the energy of the continuous-time signal? What if the continuous-time signal is known to be properly bandlimited?
- 2-6. Given a discrete-time system with impulse response  $h_k = \delta_k + \delta_{k-1}$ , what is its transfer function and frequency response? If the input is  $x_k = \cos(\omega_0 kT)$  what is the output? Show that the system has a phase response that is piecewise linear in frequency  $\omega_0$ .
- 2-7. Show that the phase response  $\theta(\omega) = \arg(H(j\omega))$  of a real system is anti-symmetric.
- 2-8. What is the impulse response of a real system that produces a constant phase shift of  $\theta$  and unity gain at all frequencies? Such a system is called a *phase shifter*.
- 2-9. Find the Fourier transform of

$$x(t) = \sum_{m=-\infty}^{\infty} \frac{1}{j(t-mT) + \alpha}.$$

- 2-10. Show that the output of an LTI system cannot contain frequencies not present in the input.
- 2-11. Sketch both a QAM modulator and demodulator for two information-bearing signals  $a(t)$  and  $b(t)$ , where your sketch includes real-valued signals only.
- 2-12.
- (a) Find an way to implement a general demodulator without using a phase splitter. Hint: You will need a lowpass filter.

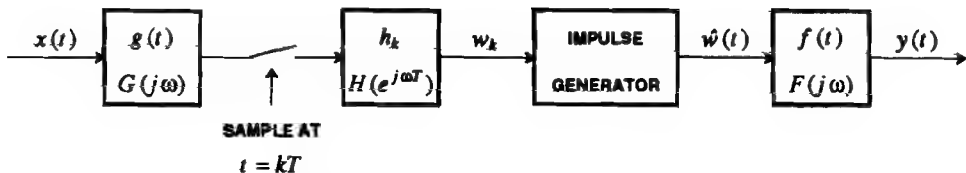


Figure 2-19. A mixed continuous and discrete-time system.

- (b) Repeat Problem 2-11 for this demodulator representation.

2-13.

- (a) Using (2.7), show that for any complex number  $z$ , the sequence  $z^k$  is an eigenfunction of a discrete-time LTI system. That is, the response to this signal is

$$y_k = H(z)z^k. \quad (2.98)$$

- (b) How is this related to the frequency response result discussed in Problem 2-2, part c?

2-14. Repeat Problem 2-13 using only the definition of a discrete-time LTI system and not using the convolution sum. (Hint: Note that  $z^{k+m} = z^k z^m$ .)

2-15. Calculate the Z transform of

$$x_k = a^k u_k, \quad u_k = \begin{cases} 1; & k \geq 0 \\ 0; & k < 0 \end{cases}. \quad (2.99)$$

where  $u_k$  is the *unit step* function.

2-16. Show that for any complex number  $z$ ,  $z^t$  is an eigenfunction of any continuous-time LTI system. Also show that for any  $z$  there exists an  $s$  such that  $e^{st} = z^t$ . Relate the eigenvalue of the system for a fixed  $z$  to the *Laplace transform*

$$H(s) = \int_{-\infty}^{\infty} e^{-st} h(t) dt. \quad (2.100)$$

The Fourier transform is the Laplace transform evaluated at  $s = j\omega$ , which explains the notation  $H(j\omega)$  used in this book.

2-17.

- (a) Show that the signals

$$x_k = \begin{cases} a^k, & k \geq 0 \\ 0, & k < 0 \end{cases} \quad y_k = \begin{cases} -a^k, & k < 0 \\ 0, & k \geq 0 \end{cases} \quad (2.101)$$

have the same Z transform.

- (b) What are the ROC for the two cases?

- (c) Under what conditions are the two signals stable? Relate this to the ROC.

2-18. Let

$$X(z) = \frac{z}{z - a} \quad (2.102)$$

and find the time domain signals for both possible ROC. Do this directly without using the results of Problem 2-17.

2-19. Given

$$X(z) = \frac{z^2}{z^2 - (a + b)z + ab} \quad (2.103)$$

where  $|a| < 1 < |b|$ , find the corresponding time-domain signal for the following two cases:

- (a) The time domain signal is known to be causal.  
 (b) The time domain signal is known to be neither causal nor anti-causal.  
 (c) Comment on whether the signal is stable in each case, and state your reasons.

2-20. Show that when the transfer function  $H(z)$  given in (2.33) has real-valued coefficients, the zeros and poles are always either real valued or come in complex-conjugate pairs.



- 2-21. Given a transfer function in the middle form of (2.33) with  $r = 0$ ,  $A = 1$ , zeros at  $1.5e^{\pm j\pi/4}$  and  $\pm j$ , and poles at  $0.5e^{\pm j\pi/8}$ . Find all the terms in the factorization (2.44). Write them in terms of polynomials with *real-valued* coefficients.

2-22.

- (a) Let  $h_k$  be a causal strictly minimum-phase sequence with a rational Z transform, and let  $g_k$  be another causal sequence obtained by taking a zero of  $H(z)$  at  $c$  and replacing it with a zero at  $1/c^*$ . Show that  $|H(e^{j\omega T})| = |G(e^{j\omega T})|$ . Hint: Find a transfer function  $A(z)$  that when multiplied by  $H(z)$  yields  $G(z)$ .
- (b) Show that

$$\sum_{k=0}^N |h_k|^2 \geq \sum_{k=0}^N |g_k|^2 \quad (2.104)$$

for all  $N \geq 0$ . Hint: Define  $F(z) = H(z)/(1 - cz^{-1})$  and write  $g_k$  and  $h_k$  in terms of  $f_k$ .

- (c) Show that for any two rational transfer functions  $H(z)$  and  $G(z)$  such that  $H(z)$  is minimum phase and  $|H(e^{j\omega T})| = |G(e^{j\omega T})|$ , (2.104) is true for all  $N \geq 0$ .

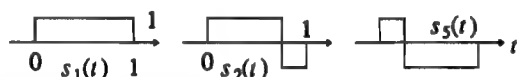
Thus, among all sequences with the same magnitude response, minimum-phase sequences are maximally concentrated near  $k = 0$  in the mean-square sense. From Parseval's formula plus the unit magnitude of an allpass filter, clearly both sides of (2.104) approach one another as  $N \rightarrow \infty$ .

- 2-23. Pass a causal input signal  $x_k$  through a first-order stable causal allpass filter such as that in Example 2-9 to yield a causal output signal  $y_k$ . Show that for any  $N \geq 0$

$$\sum_{k=0}^N |x_k|^2 \geq \sum_{k=0}^N |y_k|^2 \quad (2.105)$$

and hence the allpass filter is dispersive in the sense that it reduces the signal energy in the first  $N$  samples while keeping the total signal energy the same (since it has unit magnitude frequency response). Hint: Consider a solution method similar to Problem 2-22.

- 2-24. Use (2.44) and Example 2-11 to derive the factorization in (2.43).
- 2-25. What is the frequency response of the matched filter in Figure 2-16?
- 2-26. Given three signals  $S_1$ ,  $S_2$ , and  $S_5$ :



- (a) Find the norm of  $S_1$  and  $S_2$  and the inner product of these two signals in signal space. What is the angle between the two signals?
- (b) Find the norm of the signal  $S_1 + S_2$ .
- (c) Find a signal  $S_3$  that is orthogonal to both  $S_1$  and  $S_2$ .
- (d) Find a signal  $S_4$  that is in the subspace spanned by  $S_1$  and  $S_2$  and is orthogonal to  $S_1$ .
- (e) Find the signal in the subspace spanned by  $S_1$  and  $S_2$  that is closest to  $S_5$ .
- 2-27. Consider the space of all finite-energy continuous-time signals that are bandlimited to  $W$  radians/sec.
- (a) Show that this set of signals  $B$  is a subspace of signal space.
- (b) Characterize the subspace consisting of all signals orthogonal to every signal in  $B$ .
- (c) Find the projection of the signal  $S_1$  in Problem 2-26 on  $B$  for  $W = 1$ .
- 2-28. Given two subspaces  $M_1$  and  $M_2$  of a Hilbert space, they are orthogonal if every vector in  $M_1$  is orthogonal to every vector in  $M_2$ . The sum of the two subspaces  $M_1 \oplus M_2$  is the subspace consisting of vectors that are the sum of a vector in  $M_1$  and a vector in  $M_2$ . Given two orthogonal subspaces  $M_1$  and  $M_2$  of a Hilbert space  $H$  and an arbitrary vector  $X$  in  $H$ , show that the

projection of  $X$  on  $M_1 \oplus M_2$  can be expressed uniquely as

$$P_{M_1 \oplus M_2}(X) = P_{M_1}(X) + P_{M_2}(X) , \quad (2.106)$$

or in words the sum of the projection on  $M_1$  and the projection on  $M_2$ .

- 2-29. Given a transmitted pulse  $h(t)$  it is useful to define an *autocorrelation function*

$$\rho_k(k) = \int_{-\infty}^{\infty} h(t) h^*(t-kT) dt . \quad (2.107)$$

Show that

$$|\rho_k(k)| \leq \rho_k(0) , \quad (2.108)$$

or in words, the autocorrelation function of a pulse can never be larger than the energy of the pulse.

- 2-30. Show that (2.95) is equivalent to (2.94).

## REFERENCES

1. A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Science*, Holt, Rinehart and Winston, Inc., New York (1971).
2. C. D. McGillem and G. R. Cooper, *Continuous and Discrete Signal and System Analysis*, Holt, Rinehart, and Winston (1984).
3. A. V. Oppenheim, A. S. Willsky, and Ian T. Young, *Signals and Systems*, Prentice Hall (1983).
4. R. E. Ziemer, W. H. Tranter, and D. R. Fannin, *Signals and Systems: Continuous and Discrete*, Macmillan Publishing Co., NY (1983).
5. A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Inc. (1989).
6. L. Jackson, *Digital Filters and Signal Processing*, Kluwer Academic Publishers, Boston, MA (1985).
7. R. J. Schwarz and B. Friedland, *Linear Systems*, McGraw-Hill Book Co. (1965).
8. A. Papoulis, *The Fourier Integral and its Applications*, McGraw-Hill Book Co., New York (1962).
9. R. N. Bracewell, *The Fourier Transform and its Applications*, McGraw-Hill Book Co., New York (1965).

# 3

---

## STOCHASTIC SIGNAL PROCESSING

---

Although modulation and demodulation are deterministic, the information to be transmitted over a communication system, as well as the noise encountered in the physical transmission medium, is random or stochastic. These phenomena cannot be predicted in advance, but they have certain predictable characteristics which can be summarized in a random process model. The design of a digital communication system heavily exploits these characteristics.

In this chapter we review the notation that will be used for random variables and processes, and cover several topics in detail that may be new to some readers and are particularly important in the sequel. These include Chernoff bounding techniques, Bayes' rule, and mixtures of discrete-time and continuous-time random processes. Markov chains are discussed in Section 3.3, and will be used in a diverse set of applications in Chapters 9, 10, 12-14, and 19. Section 3.4, on Poisson processes, uses the Markov chain results to describe Poisson processes and shot noise, which will be important to the understanding of optical fiber systems in Chapters 5 and 8.

### 3.1. RANDOM VARIABLES

Before reviewing the theory of the stochastic process, we review some theory and notation associated with random variables. In digital communication it is common to encounter combinations of discrete and continuous-valued random variables,

so this will be emphasized.

We denote a *random variable* by a capital letter, such as  $X$ , and an *outcome* of the random variable by a lower-case letter, such as  $x$ . The random variable is a real or complex-valued function defined on the *sample space*  $\Omega$  of all possible outcomes. An *event*  $E$  is a set of possible outcomes and is assigned a probability, written  $\Pr[E]$ , where  $0 \leq \Pr[E] \leq 1$ . Since an event is a set, we can define the union of two events,  $E_1 \cup E_2$  or the intersection of events  $E_1 \cap E_2$ . The basic formula

$$\Pr[E_1 \cup E_2] = \Pr[E_1] + \Pr[E_2] - \Pr[E_1 \cap E_2] \quad (3.1)$$

leads to the very useful *union bound*,

$$\Pr[E_1 \cup E_2] \leq \Pr[E_1] + \Pr[E_2]. \quad (3.2)$$

The *cumulative distribution function (c.d.f.)* of a real valued random variable  $X$  is the probability of the event  $X \leq x$ ,

$$F_X(x) = \Pr[X \leq x]. \quad (3.3)$$

Where there can be no confusion, we often omit the subscript, writing the c.d.f. as  $F(x)$ . For a complex-valued random variable  $Y$ ,

$$F_Y(y) = \Pr[\operatorname{Re}\{Y\} \leq \operatorname{Re}\{y\}, \operatorname{Im}\{Y\} \leq \operatorname{Im}\{y\}]. \quad (3.4)$$

For a continuous real-valued random variable, the *probability density function (p.d.f.)*  $f_X(x)$  is defined such that for any interval  $I \subset \mathbf{R}$

$$\Pr[X \in I] = \int_I f_X(x) dx. \quad (3.5)$$

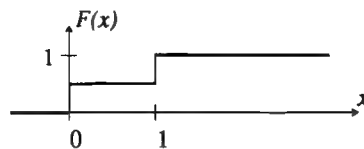
For a complex-valued random variable,  $I$  is a region in the complex plane. For a real-valued random variable  $X$ ,

$$f_X(x) = \frac{d}{dx} F_X(x), \quad (3.6)$$

where the derivative exists. We will often use the generalized derivative, so that when the c.d.f. includes a step function the corresponding p.d.f. has a Dirac delta function.

### Example 3-1.

For the c.d.f. shown below,



the p.d.f. consists exclusively of Dirac delta functions,

$$f(x) = 0.5 \cdot \delta(x) + 0.5 \cdot \delta(x-1). \quad (3.7)$$

Such a density is characteristic of a discrete random variable.  $\square$

For a discrete-valued random variable  $X$ , we will denote the probability of an outcome  $x \in \Omega$  as

$$p_X(x) = \Pr[X = x], \quad (3.8)$$

where we will again omit the subscript where there can be no confusion. The p.d.f. can be written as

$$f_X(x) = \sum_{y \in \Omega_X} p_X(y) \delta(x - y). \quad (3.9)$$

The *expected value* or *mean* of  $X$  is defined as

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx \quad \text{or} \quad E[X] = \sum_{x \in \Omega} x \cdot p_X(x), \quad (3.10)$$

for continuous-valued and discrete-valued random variables, respectively. For a complex-valued random variable  $Y$ , we integrate over the complex plane,

$$E[Y] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x + iz) f_Y(x + iz) dx dz. \quad (3.11)$$

The *fundamental theorem of expectation* states that if  $g(\cdot)$  is any function defined on the sample space of  $X$ , then

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx. \quad (3.12)$$

Especially important expectations are the mean and variance, defined as

$$\mu = E[X], \quad \sigma_X^2 = E[(X - E[X])^2] = E[X^2] - E[X]^2. \quad (3.13)$$

For complex-valued random variables, the variance is defined similarly as

$$\sigma_X^2 = E[|X|^2] - |E[X]|^2 = E[XX^*] - E[X] \{E[X]\}^*. \quad (3.14)$$

The *joint c.d.f.* of two real-valued random variables  $X$  and  $Y$  is

$$F_{X,Y}(x,y) = \Pr(X \leq x, Y \leq y) = \int_{-\infty}^x \int_{-\infty}^y f_{X,Y}(\alpha, \beta) d\alpha d\beta, \quad (3.15)$$

where  $f_{X,Y}(x,y)$  is the *joint p.d.f.* The joint p.d.f. can be written in terms of the joint c.d.f. as

$$f(x,y) = \frac{\partial^2}{\partial x \partial y} F(x,y), \quad (3.16)$$

where we have omitted the subscripts as before. The *marginal density*  $f_X(x)$  of a random variable  $X$  can be found from the joint p.d.f. from

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y) dy. \quad (3.17)$$

The random variables  $X$  and  $Y$  are *independent* or *statistically independent* if for all intervals  $I$  and  $J$ ,

$$\Pr[X \in I \cap Y \in J] = \Pr[X \in I] \Pr[Y \in J] , \quad (3.18)$$

which is equivalent to

$$f_{X,Y}(x,y) = f_X(x)f_Y(y) \quad \text{or} \quad F_{X,Y}(x,y) = F_X(x)F_Y(y) . \quad (3.19)$$

Independence implies that the *cross-correlation* is

$$E[XY] = E[X]E[Y] . \quad (3.20)$$

When (3.20) is satisfied, the random variables are said to be *uncorrelated*. Two random variables can be uncorrelated and yet not be independent.

### 3.1.1. Moment Generating Function and Chernoff Bound

The *characteristic function* of  $X$  is defined as

$$\Phi_X(s) = E[e^{sX}] = \int_{-\infty}^{\infty} e^{sx} f_X(x) dx \quad (3.21)$$

for a complex variable  $s$ . This is the Laplace transform of  $f_X(x)$  evaluated at  $x = -s$ . When  $s$  is real-valued, which will suffice for applications in this book, (3.21) is called the *moment generating function*.

#### Exercise 3-1.

Define  $Z = X + Y$  where  $X$  and  $Y$  are independent, and show that

$$\Phi_Z(s) = \Phi_X(s)\Phi_Y(s) . \quad (3.22)$$

□

#### Exercise 3-2.

Show that

$$E[X] = \frac{\partial}{\partial s} \Phi_X(s) \Big|_{s=0} , \quad E[X^2] = \frac{\partial^2}{\partial s^2} \Phi_X(s) \Big|_{s=0} . \quad (3.23)$$

□

The *Chernoff bound*, based on the moment generating function, is very useful for bounding the tail probability for a random variable where an exact evaluation is intractable.

#### Exercise 3-3.

(a) Show that the probability of event  $X > x$  is bounded by

$$1 - F_X(x) = \Pr[X > x] \leq e^{-sx} \Phi_X(s) \quad (3.24)$$

for any real-valued  $s \geq 0$ . This establishes that the tail of the p.d.f. decreases at least exponentially for any distribution for which the moment generating function exists. (Hint: Write the probability as the integral against a step function, and bound the step function by an exponential.)

(b) Find the similar bound

$$F_X(x) \leq e^{sx} \Phi_X(-s) \quad (3.25)$$

for  $s \geq 0$ .

(c) Show that the  $s$  that minimizes the bound (makes it tightest) in (a) and (b) must satisfy

$$x \Phi_X(s) = \frac{\partial \Phi_X(s)}{\partial s}, \quad -x \Phi_X(-s) = \frac{\partial \Phi_X(-s)}{\partial s}, \quad (3.26)$$

respectively.  $\square$

### 3.1.2. Conditional Probabilities and Bayes' Rule

The *conditional probability* that a continuous-valued random variable  $X$  is in the interval  $I$  given that  $Y$  is in the interval  $J$  is defined for all  $J$  such that  $\Pr[Y \in J] \neq 0$  to be

$$\Pr[X \in I | Y \in J] = \frac{\Pr[X \in I \cap Y \in J]}{\Pr[Y \in J]}, \quad (3.27)$$

where  $\Pr[Y \in J]$  is called a *marginal probability* because it does not consider the possible effects of  $X$  on  $Y$ . For complex or vector-valued random variables,  $I$  and  $J$  are regions or volumes, rather than intervals. If  $X$  and  $Y$  are independent, then  $\Pr[X \in I | Y \in J] = \Pr[X \in I]$ . The joint probability can be written in terms of the conditional probabilities,

$$\Pr[X \in I \cap Y \in J] = \Pr[X \in I | Y \in J] \Pr[Y \in J]. \quad (3.28)$$

Equivalently,

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}, \quad (3.29)$$

where  $f_Y(y)$  is called a *marginal density*.

The *conditional density*  $f_{X|Y}(x|y)$  is well defined only for  $y$  such that  $f_Y(y) \neq 0$ .

Since  $f_{X,Y}(x,y) = f_{Y,X}(y,x)$ , (3.29) implies that

$$f_{X|Y}(x|y)f_Y(y) = f_{Y|X}(y|x)f_X(x), \quad (3.30)$$

which is a form of *Bayes' rule*.

It is common in digital communication systems to encounter both discrete-valued and continuous-valued random variables in the same system. In this case, (3.30) has Dirac delta functions.

#### Exercise 3-4.

Suppose that  $Y$  is discrete-valued and  $X$  is continuous-valued. Show that by integrating (3.30) over small intervals about  $y$ , we get the mixed form of Bayes' rule,

$$f_{X|Y}(x|y)p_Y(y) = p_{Y|X}(y|x)f_X(x). \quad (3.31)$$

This involves both probabilities and probability density functions. It has no delta functions as long as  $X$  is continuous-valued. If  $X$  is also discrete-valued, show that then

$$p_{X|Y}(x|y)p_Y(y) = p_{Y|X}(y|x)p_X(x), \quad (3.32)$$

which has only discrete probabilities.  $\square$

For discrete-valued distributions, the marginal probability can be written in terms of the conditional probabilities as

$$p_Y(y) = \sum_{x \in \Omega} p_{Y|X}(y|x)p_X(x) = \sum_{x \in \Omega} p_{Y,X}(y, x), \quad (3.33)$$

where  $\Omega$  is the countable sample space for  $X$ . This relation shows us how to obtain the marginal probabilities of a random variable given only joint probabilities, or given only conditional probabilities and the marginal probabilities of the other random variable. Using this relation, we can write the conditional probability of  $X$  given  $Y$  in terms of the conditional probability of  $Y$  given  $X$  and the marginal probability of  $X$

$$p_{X|Y}(x|y) = \frac{p_{Y|X}(y|x)p_X(x)}{\sum_{x \in \Omega_X} p_{Y|X}(y|x)p_X(x)}. \quad (3.34)$$

This relation is known as *Bayes' theorem*. The analogous Bayes' theorem for continuous-valued random variables is

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)f_X(x)}{\int_{x \in \Omega_X} f_{Y|X}(y|x)f_X(x) dx}. \quad (3.35)$$

### 3.1.3. Gaussian Random Variables and the Central Limit Theorem

A *Gaussian* or *normal* random variable has the p.d.f.

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \quad (3.36)$$

where  $\sigma^2$  is the variance and  $\mu$  is the mean. The c.d.f. can be expressed only as an integral,

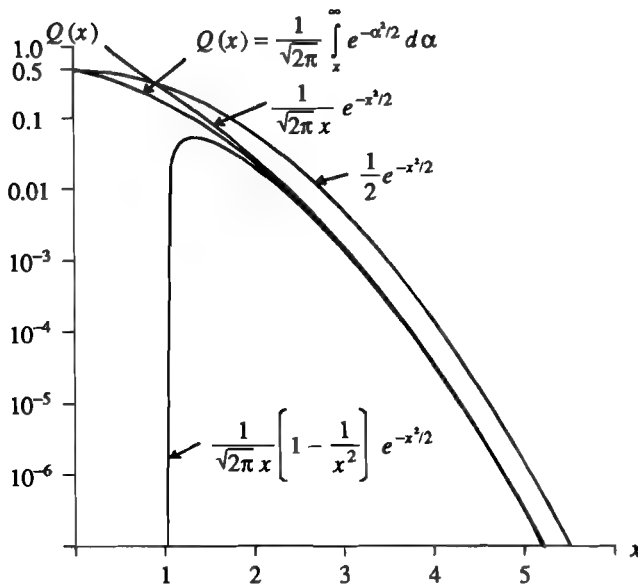
$$F_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-(\alpha-\mu)^2/2\sigma^2} d\alpha \quad (3.37)$$

for which there is no closed-form expression. The *standard Gaussian* random variable is a zero-mean Gaussian random variable  $X$  with variance  $\sigma^2 = 1$ . The *complementary distribution function* of this standard Gaussian is denoted by the special notation  $Q(x)$ ,

$$Q(x) = \Pr[X > x] = 1 - F_X(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\alpha^2/2} d\alpha. \quad (3.38)$$

$Q(x)$ , therefore, is the integral of the tail of the Gaussian density. It is plotted in Figure 3-1 using a log scale for probability. The function is related to the well-tabulated *error function* ( $\text{erf}(x)$ ) and the *complementary error function* ( $\text{erfc}(x)$ ) by





**Figure 3-1.** The probability  $Q(x)$  that a zero-mean, unit-variance Gaussian random variable  $X$  (the standard Gaussian) exceeds  $x$ , plotted on a log scale.

$$Q(x) = \frac{1}{2} \operatorname{erfc} \left[ \frac{x}{\sqrt{2}} \right] = \frac{1}{2} \left[ 1 - \operatorname{erf} \left[ \frac{x}{\sqrt{2}} \right] \right]. \quad (3.39)$$

**Exercise 3-5.**

Show that for a Gaussian random variable  $X$  with mean  $\mu$  and variance  $\sigma^2$ ,

$$\Pr[X > x] = Q \left[ \frac{x - \mu}{\sigma} \right]. \quad (3.40)$$

□

Although  $Q(\cdot)$  can only be tabulated or numerically determined, a useful bound follows from the Chernoff bound of Exercise 3-3.

**Exercise 3-6.**

- (a) Show that the moment generating function of a Gaussian random variable with mean  $\mu$  and variance  $\sigma^2$  is

$$\log_e \Phi_X(s) = \mu s + \sigma^2 s^2 / 2. \quad (3.41)$$

- (b) Show from the Chernoff bound (Exercise 3-3) that

$$1 - F_X(x) \leq e^{-(x - \mu)^2 / 2\sigma^2} \quad (3.42)$$

and thus that

$$Q(x) \leq e^{-x^2/2}. \quad (3.43)$$

□

Tighter bounds are derived in Problem 3-3 and plotted in Figure 3-1.

Use of the Gaussian distribution for modeling noise phenomena can be justified on physical grounds by the *central limit theorem*. It states, roughly, that the Gaussian distribution is a good model for the cumulative effect of a large number of independent random variables, regardless of the nature of their individual distributions. More precisely, let  $\{Y_i\}$  for  $1 \leq i \leq N$  denote a set of  $N$  statistically independent zero-mean random variables, each with the same p.d.f.  $f_{Y_i}(y) = f(y)$  and finite variance  $\sigma^2$ . That is, the random variables are *independent and identically distributed* (i.i.d.). Define a random variable  $Z$  that is a normalized sum of the  $Y_i$ ,

$$Z = \frac{1}{\sqrt{N}} \sum_{i=1}^N Y_i. \quad (3.44)$$

Then the distribution function of  $Z$  approaches Gaussian,  $(1 - Q(z/\sigma))$ , as  $N \rightarrow \infty$ . If each random variable  $Y_i$  represents some individual physical phenomenon, and  $Z$  is the cumulative effect of these phenomena, then as  $N$  gets large, the distribution of  $Z$  becomes Gaussian, regardless of the distribution of each  $Y_i$ .

In view of this theorem, it is hardly surprising that the sum of independent Gaussian random variables is Gaussian.

#### Exercise 3-7.

For an arbitrary linear combination of  $N$  zero mean independent Gaussian random variables  $X_i$ , each with variance  $\sigma^2$ ,

$$Z = a_1 X_1 + \cdots + a_N X_N, \quad (3.45)$$

use the moment generating function to show that  $Z$  is itself zero-mean Gaussian with variance

$$\sigma_Z^2 = (a_1^2 + \cdots + a_N^2) \sigma^2. \quad (3.46)$$

□

Two zero-mean Gaussian random variables with variance  $\sigma^2$  are *jointly Gaussian* if their joint p.d.f. is

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \exp \left[ -\frac{x^2 - 2\rho xy + y^2}{2\sigma^2(1-\rho^2)} \right], \quad (3.47)$$

where  $\rho$  is called the *correlation coefficient*,

$$\rho = \frac{E[XY]}{\sigma^2}. \quad (3.48)$$

Note that  $-1 \leq \rho \leq 1$ , and if  $X$  and  $Y$  are uncorrelated then  $\rho = 0$ .

### Exercise 3-8.

Show that two jointly Gaussian random variables are statistically independent if and only if they are uncorrelated.  $\square$

This definition can be extended to  $N > 2$  jointly Gaussian random variables. If a random vector  $\mathbf{X}$  has components that are jointly zero-mean independent Gaussian random variables with the same variance  $\sigma^2$ , then the joint p.d.f. is

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{M/2} \sigma^M} \exp \left[ -\frac{1}{2\sigma^2} \|\mathbf{x}\|^2 \right], \quad (3.49)$$

where  $M$  is the number of components in the vector and  $\|\mathbf{x}\|$  is the Euclidean norm (2.73) of the vector. When  $\mathbf{X}$  is complex-valued with independent real and imaginary parts, (3.49) still holds. Any linear combination of jointly Gaussian random variables is Gaussian (as we saw in Exercise 3-7 for independent zero-mean Gaussian random variables).

This can be further generalized. A vector  $\mathbf{X}$  with  $M$  jointly Gaussian real-valued random variables has p.d.f.

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{M/2} |\mathbf{C}_{\mathbf{X}}|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \mathbf{m}_{\mathbf{X}})^T \mathbf{C}_{\mathbf{X}}^{-1} (\mathbf{x} - \mathbf{m}_{\mathbf{X}}) \right], \quad (3.50)$$

where

$$\mathbf{C}_{\mathbf{X}} = E[(\mathbf{x} - \mathbf{m}_{\mathbf{X}})(\mathbf{x} - \mathbf{m}_{\mathbf{X}})^T] \quad (3.51)$$

is the *covariance matrix*,  $|\mathbf{C}_{\mathbf{X}}|$  is its determinant, and  $\mathbf{m}_{\mathbf{X}} = E[\mathbf{X}]$  is the vector mean. In the special case that the vector mean is zero and elements of the random vector are independent with equal variances,  $\mathbf{C}_{\mathbf{X}}$  becomes diagonal and (3.50) reduces to (3.49). An important observation from (3.50) is that the p.d.f. of a Gaussian random vector is completely specified by the vector mean and the pairwise covariances contained in the covariance matrix. Consequently, these two sets of parameters completely specify all the statistical properties of a Gaussian random vector.

### 3.1.4. Geometric Interpretation

Random variables can be interpreted geometrically using the approach of Section 2.6. In particular, consider the set of all complex-valued random variables  $X$  with bounded second moments,  $E[|X|^2] < \infty$ , and associate a vector  $\mathbf{X}$  with each random variable,

$$\mathbf{X} \leftrightarrow X. \quad (3.52)$$

### Exercise 3-9.

Make reasonable definitions for the operations of addition of vectors, multiplication by a scalar, the vector 0, and the additive inverse. Show that the set of such vectors form a linear space (Section 2.6.1).  $\square$

An inner product on this space can be defined as

$$\langle X, Y \rangle = E[XY^*]. \quad (3.53)$$

**Exercise 3-10.**

Show that (3.53) is a legitimate inner product (Section 2.6.2).  $\square$

This geometric interpretation pays dividends in understanding the results of linear prediction theory (Section 3.2.3).

## 3.2. RANDOM PROCESSES

A discrete-time random process  $\{X_k\}$  is a sequence of random variables indexed by integers  $k$ , while a continuous-time random process  $X(t)$  is indexed by a real variable  $t$ . We write an *outcome* of  $\{X_k\}$  or  $\{X(t)\}$  as the lower case deterministic signal  $\{x_k\}$  or  $\{x(t)\}$ . When there can be no confusion between a *signal* and a *sample of the signal*, we omit the braces  $\{\cdot\}$ . Each random sample  $X_k$  or  $X(t)$  may be complex, vector-valued, or real-valued.

**Example 3-2.**

A real-valued random process  $X(t)$  is a *Gaussian random process* if its samples  $\{X(t_1), \dots, X(t_N)\}$  are jointly Gaussian random variables for any  $N$  and for any  $\{t_1, \dots, t_N\}$ .  $\square$

The first and second moments of the random process are the *mean*

$$m_k = E[X_k], \quad m(t) = E[X(t)] \quad (3.54)$$

and the *autocorrelation*

$$R_{XX}(k, i) = E[X_k X_i^*], \quad R_{XX}(t_1, t_2) = E[X(t_1)X^*(t_2)]. \quad (3.55)$$

where  $X^*$  is the complex conjugate of  $X$ .

**Example 3-3.**

Consider a real-valued, zero-mean Gaussian random process. A random vector  $\mathbf{X}$  can be constructed from some arbitrary set of samples. For such a vector, the covariance matrix of (3.51) can be obtained from the autocorrelation function (3.55). Consequently, the joint p.d.f. (3.50) of any set of samples can be obtained from the autocorrelation function. Thus, the statistical properties of a zero-mean real-valued Gaussian random process are completely specified by its autocorrelation function.  $\square$

A random process is *strict-sense stationary* if the p.d.f. for any sample is independent of the time index of the sample, and the joint p.d.f. of any set of samples depends only on the time differences between samples, and not on the absolute time of any sample. It is *wide-sense stationary (WSS)* if its mean is independent of the time

index, and its autocorrelation depends only on the time difference between samples, and not on the absolute time. In other words,  $m_k$  or  $m(t)$  must be constant and  $R_{XX}(k, i)$  or  $R_{XX}(t_1, t_2)$  must be a function only of the difference  $k - i$  or  $t_1 - t_2$ . Strict sense stationarity implies wide-sense stationarity, but not the reverse, unless the process is Gaussian.

### Example 3-4.

A real-valued WSS Gaussian random process is also strict-sense stationary. The autocorrelation function and mean of such a process can be used to construct the covariance matrix (3.51) for any set of samples. Since the process is WSS, the entries in the matrix will be independent of the absolute time index of the samples, and will depend instead only on the time differences between samples. Consequently, the joint p.d.f. (3.50) of any set of samples will depend only on these time differences. Hence the process is strict-sense stationary.  $\square$

For a WSS random process the autocorrelation function can be written in terms of the time difference between samples,  $m = k - i$  or  $\tau = t_1 - t_2$ , yielding the simpler notation

$$R_X(m) = E[X_{k+m} X_k^*], \quad R_X(\tau) = E[X(t + \tau) X^*(t)]. \quad (3.56)$$

$R_X(0)$  is the second moment of the samples

$$R_X(0) = E[|X_k|^2], \quad R_X(0) = E[|X(t)|^2], \quad (3.57)$$

and can be interpreted as the *power* of a random process. For a WSS random process, the *power spectral density* or *power spectrum* is the Fourier transform of the autocorrelation function,

$$S_X(e^{j\omega T}) = \sum_{m=-\infty}^{\infty} R_X(m) e^{-j\omega m T}, \quad S_X(j\omega) = \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega \tau} d\tau, \quad (3.58)$$

where  $T$  is the sample interval of the discrete-time random process. The power therefore is the integral of the power spectrum,

$$R_X(0) = \frac{T}{2\pi} \int_{-\pi/T}^{+\pi/T} S_X(e^{j\omega T}) d\omega, \quad R_X(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(j\omega) d\omega. \quad (3.59)$$

The power spectrum is real-valued since the autocorrelation function is conjugate symmetric,  $R_X(m) = R_X^*(-m)$  or  $R_X(\tau) = R_X^*(-\tau)$ . It is also non-negative (see Problem 3-9). Furthermore, if  $X_k$  or  $X(t)$  is real-valued, then the power spectrum is symmetric about  $\omega = 0$ . We can also write the power spectrum as a Z transform or a Laplace transform,

$$S_X(z) = \sum_{m=-\infty}^{\infty} R_X(m) z^{-m}, \quad S_X(s) = \int_{-\infty}^{\infty} R_X(\tau) e^{-s\tau} d\tau. \quad (3.60)$$

Evaluating  $S_X(z)$  on the unit circle or  $S_X(s)$  on the  $j\omega$  axis yields (3.58).

**Example 3-5.**

Consider a zero-mean random process  $\{X_k\}$  where the samples  $X_k$  are all independent and identically distributed (i.i.d.) zero-mean random variables with variance  $\sigma_x^2$ . In this case  $R_X(k) = \sigma_x^2 \delta_k$  and the power spectrum is a constant,  $S_X(e^{j\omega T}) = \sigma_x^2$ , independent of the frequency, with power  $R_X(0) = \sigma_x^2$ .  $\square$

Any zero-mean process with a constant power spectrum is said to be a *white random process*. This may or may not imply that the samples of the random process are independent, although for the important Gaussian case they are.

**Example 3-6.**

As in Example 3-5, consider a continuous-time random process  $\{X(t)\}$  with the autocorrelation function

$$R_X(\tau) = N_0 \delta(\tau). \quad (3.61)$$

The power spectrum of this process is a constant,  $S_X(j\omega) = N_0$ , so  $\{X(t)\}$  is white. The power  $R_X(0)$  of this continuous-time white process is infinite. So we immediately run into mathematical difficulties for the continuous-time case that we did not encounter in the discrete-time case.  $\square$

Although the continuous-time white random process of Example 3-6 leads to the non-physical condition that the power is infinite (or undefined), it is an extremely important model. It would appear from the fact that  $R_X(\tau) = 0$  for all  $\tau \neq 0$  that any two distinct samples of a continuous-time white random process are uncorrelated, but, unfortunately, this makes no mathematical or physical sense. Sampling a continuous-time white random process is an ill-defined concept. Roughly speaking, a continuous-time white random process varies so quickly that it is not possible to determine its characteristics at any instant in time.

In spite of these mathematical difficulties, the continuous-time white random processes is useful as a model for noise which has an approximately constant power spectrum over a bandwidth larger than the bandwidth of the system we are considering. In such a system we will always bandlimit the noise to eliminate any out-of-band component. In this event, it makes no difference if we start with a white noise or a more accurate model; the result will be very nearly the same. But using the white noise model results in significantly simpler algebraic manipulation. In this book we will often use the white noise model, and take care to always bandlimit this noise process prior to other operations such as sampling. After bandlimiting, we obtain a well-behaved process with finite power.

**Example 3-7.**

*Thermal* or *Johnson* noise in electrical resistors has a power spectrum that is flat to more than  $10^{12}$  Hz, a bandwidth much greater than most systems of interest (see [1]). Thus, we can safely use white noise as a model for this thermal noise without compromising accuracy. The noise in the model at frequencies greater than  $10^{12}$  Hz will always be filtered out at the input to our system anyway. By contrast, in optical systems (Section 5.3), thermal noise is generally insignificant at optical frequencies. Thermal noise is modeled as a Gaussian random process, from the central limit theorem, since it is comprised of the superposition of many independent events (thermal fluctuations of individual electrons).  $\square$

### 3.2.1. Cross-Correlation and Complex Processes

Given two random processes  $X(t)$  and  $Y(t)$ , we can define a *cross-correlation* function,

$$R_{XY}(t_1, t_2) = E[X(t_1)Y^*(t_2)] . \quad (3.62)$$

If  $X(t)$  and  $Y(t)$  are each wide-sense stationary, then they are *jointly wide-sense stationary* if  $R_{XY}(t_1, t_2)$  is a function only of  $(t_1 - t_2)$ .

A complex-valued random process  $X(t)$  is defined as

$$X(t) = \text{Re}\{X(t)\} + j \cdot \text{Im}\{X(t)\} , \quad (3.63)$$

where  $\text{Re}\{X(t)\}$  and  $\text{Im}\{X(t)\}$  are real-valued random processes. The second order statistics of such a process consist of the two autocorrelation functions of the real and imaginary parts, as well as their cross-correlation functions. Complex Gaussian random processes are very important in digital communication systems; they have some special properties that are considered in detail in Chapter 8.

### 3.2.2. Filtered Random Processes

A particular outcome  $x_k$  or  $x(t)$  of a random process is a signal, and therefore may be filtered or otherwise processed. We can also talk about filtering the random process  $X_k$  or  $X(t)$  itself, rather than an outcome. Then we get a new random process with a sample space that is obtained by applying every element of the sample space of the original random process to the input of the filter.

#### Example 3-8.

A filtered Gaussian random process is a Gaussian random process. Intuitively, this is true because filtering is linear, and any linear combination of jointly Gaussian random variables is a Gaussian random variable.  $\square$

Consider the two continuous-time LTI systems shown in Figure 3-2 with WSS continuous-time random process inputs.

#### Exercise 3-11.

Show that the output of the filter  $h(t)$  is WSS, and that its autocorrelation function and power spectrum are given by

$$R_W(\tau) = h(\tau) * h^*(-\tau) * R_X(\tau) , \quad S_W(j\omega) = S_X(j\omega) |H(j\omega)|^2 , \quad (3.64)$$

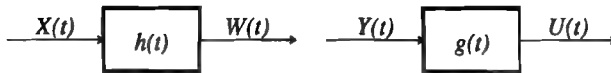


Figure 3-2. Two linear systems with WSS random process inputs.

$$S_W(s) = S_X(s)H(s)H^*(-s^*). \quad (3.65)$$

□

**Exercise 3-12.**

Show that if a WSS discrete-time random process  $X_k$  is filtered by a filter that has impulse response  $h_k$ , and the result is  $W_k$ , then  $W_k$  is WSS and

$$R_W(m) = h_m * h_{-m}^* * R_X(m), \quad S_W(e^{j\omega T}) = S_X(e^{j\omega T}) |H(e^{j\omega T})|^2, \quad (3.66)$$

$$S_W(z) = S_X(z)H(z)H^*(1/z^*). \quad (3.67)$$

□

**Example 3-9.**

A white random process  $X(t)$  has power spectrum  $S_X(j\omega) = N_0$ , a constant. If it is filtered by an ideal LPF with transfer function

$$H(j\omega) = \text{rect}(\omega, 2\pi \times 10^{12}) = \begin{cases} 1; & |\omega| < 2\pi \times 10^{12} \\ 0; & \text{otherwise} \end{cases} \quad (3.68)$$

then the power spectrum of the output is

$$S_W(j\omega) = N_0 \text{rect}(\omega, 2\pi \times 10^{12}) = \begin{cases} N_0; & |\omega| < 2\pi \times 10^{12} \\ 0; & \text{otherwise} \end{cases} \quad (3.69)$$

which is a reasonable approximation to thermal noise in a resistor. Furthermore, since thermal noise is the cumulative effect of random motion of a huge number of individual particles, we can apply the central limit theorem to argue that a sample of such thermal noise should be a Gaussian random variable. Thus we conclude that thermal noise is reasonably modeled as white Gaussian noise. □

The *cross-spectral density* of two jointly WSS random processes at the filter inputs in Figure 3-2 is defined as the Fourier transform of the cross-correlation function,

$$S_{XY}(j\omega) = \int_{-\infty}^{\infty} R_{XY}(\tau) e^{-j\omega\tau} d\tau, \quad R_{XY}(\tau) = E[X(t + \tau)Y^*(t)]. \quad (3.70)$$

**Exercise 3-13.**

Show that the cross-power spectrum of the outputs in Figure 3-2 is

$$S_{WU}(j\omega) = H(j\omega)G^*(j\omega)S_{XY}(j\omega). \quad (3.71)$$

□

**3.2.3. The Innovations Process**

Given a wide-sense stationary random process  $\{X_k\}$  with power spectrum  $S_X(e^{j\omega T})$ , a natural *innovations representation* of that random process follows from the monic minimum-phase spectral factorization of  $S_X(z)$  (see (2.55) in Section



2.5.5). In particular, since  $S_X(z)$  is real-valued and non-negative on the unit circle, it can be decomposed as

$$S_X(z) = A_x^2 G_x(z) G_x^*(1/z^*) \quad (3.72)$$

where  $G_x(z)$  is a monic loosely minimum-phase causal filter, and  $A_x^2$  is a constant to be interpreted shortly. If  $S_X(z)$  has no zeros on the unit circle ( $S_X(e^{j\omega T}) > 0$  for all  $\omega$ ), then  $G_x(z)$  is strictly minimum phase. In this case, its inverse filter  $G_x^{-1}(z)$  is stable, and is also a monic minimum-phase causal filter. If we filter the process  $X_k$  with the filter  $G_x^{-1}(z)$ , as shown in Figure 3-3, then from (3.67), the output  $I_k$  is a white random process with power spectrum  $S_I(z) = A_x^2$ . The random process  $\{I_k\}$  is called the *innovations process*. Its power is  $A_x^2$ .

The innovations process and the filter  $G_x(z)$  can be used to generate the random process  $X_k$ , as shown in Figure 3-3. This helps to explain the terminology. Since  $I_k$  is white, each new sample is uncorrelated with previous samples. Thus each new sample brings new information (an "innovation") about the random process  $X_k$ . Viewed another way, the *whitening filter*  $G^{-1}(z)$  removes redundant information from  $X_k$  by removing correlated components in the samples. What is left has only uncorrelated samples. Thus we can think of  $X_k$  as having two components; the innovation is the new or "random" part, while the remainder is a linear combination of past innovations.

### 3.2.4. Linear Prediction

A *linear predictor* forms an estimate of the current sample of a discrete-time random process from a linear combination of the past samples. It uses the correlation between samples to construct an informed estimate of the current sample based on the past.

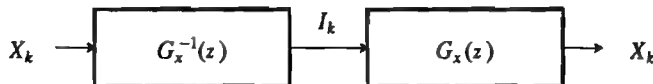
If the transfer function of the predictor is  $F(z)$ , it must be strictly causal,

$$F(z) = \sum_{k=1}^{\infty} f_k z^{-k}. \quad (3.73)$$

This ensures that only past samples are used in constructing the prediction. The *prediction error*, formed by taking the difference between the current sample and the prediction, is generated by applying a filter with transfer function

$$E(z) = 1 - F(z) \quad (3.74)$$

to the random process.  $E(z)$  must be stable, causal, and monic to be a legitimate



**Figure 3-3.** Generation of the innovations  $I_k$  from  $X_k$ , and the recovery of  $X_k$  from its innovations.

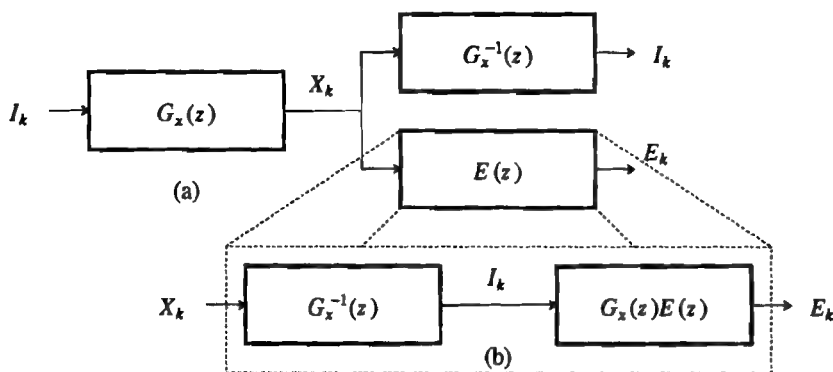
*prediction error filter.* The prediction error filter  $E(z)$  (or equivalently  $F(z)$ ) should be designed to minimize the power of the prediction error sequence  $E_k$ . We will now show that  $E(z) = G_x^{-1}(z)$  is optimal, where  $G(z)$  results from the spectral factorization in (3.72).

In Figure 3-4a we show the innovations representation of  $X_k$ , where it is generated by filtering the innovations process  $I_k$  with the filter  $G_x(z)$ . Two prototype prediction error filters are shown, a general filter  $E(z)$ , which is constrained to be causal and monic, and a specific filter  $G_x^{-1}(z)$ , which is causal and monic. We will now demonstrate that the lower output  $E_k$  cannot have less power than the upper output  $I_k$ , so the upper filter  $G_x^{-1}(z)$  is an optimal prediction error filter. The prediction error for the optimal predictor is therefore precisely the innovation  $I_k$ , and the prediction error power is  $A_x^2$ .

To show this, consider Figure 3-4b, where  $E(z)$  is split into two filters  $G_x^{-1}(z)$  and  $G_x(z)E(z)$ . Such a split might be absurd in implementation, but mathematically it is perfectly reasonable. The output of the first filter is the innovations process  $I_k$ . We will now show that the output of the second filter cannot have lower power than that of  $I_k$ . Since both  $G_x(z)$  and  $E(z)$  are causal and monic filters ( $G_x(\infty) = E(\infty) = 1$ ) it follows that  $G_x(z)E(z)$  must also be causal and monic ( $G_x(\infty)E(\infty) = 1$ ). Let this filter have impulse response  $f_k, 0 \leq k < \infty$ , where  $f_0 = 1$ . Since the input innovations process is white with variance  $A_x^2$ , the output variance is

$$A_x^2 \sum_{k=0}^{\infty} |f_k|^2 \geq A_x^2, \quad (3.75)$$

with equality if and only if  $f_k = 0, k \geq 1$ , or in other words  $E(z) = G_x^{-1}(z)$ .



**Figure 3-4.** Steps in the derivation of the optimal linear prediction error filter. (a) Comparison of two prediction error filters. (b) Decomposition of the general filter  $E(z)$ .

Intuitively, since the predictor is exploiting the correlation of input samples, we would expect the prediction error to be white, since otherwise there would still be correlation to further exploit. Thus, the second filter in Figure 3-4b is counterproductive, since it introduces correlation. However, this intuitive explanation is incomplete, because  $G_x(z)E(z)$  could have a flat frequency response, in which case  $E_k$  would still be white even though it is not the innovations process for  $X_k$ ! This case is addressed by the following exercise.

**Exercise 3-14.**

Show that if  $H(z)$  is rational, causal, and monic, and has a flat frequency response  $|H(e^{j\omega T})| = K$ , then  $K > 1$ . Thus, a monic filter with a flat frequency response must have gain larger than unity, and thus its white output has a larger variance than its input.  $\square$

This exercise is instructive, because it shows that any causal and monic filter with a flat frequency response will amplify its inputs. The optimal prediction error filter  $G_x^{-1}(z)$  thus has two key properties: it is a *whitening filter*, resulting in a white prediction error, and it is *minimum-phase*. The whitening filter property of the prediction error filter (if not the minimum-phase property) can also be demonstrated by orthogonality arguments (see Problem 3-5), and has a simple geometric interpretation (see Problem 3-6).

### 3.2.5. Sampling a Random Process

A finite power continuous-time random process  $X(t)$  can be sampled, yielding a discrete-time random process  $Y_k = X(kT)$ . Since we will be performing this sampling operation often in digital communication systems, it is important to relate the statistics of the continuous-time random processes with those of the discrete-time random process obtained by sampling it. Assuming  $X(t)$  is WSS,

$$R_{YY}(k, i) = E[X(kT)X^*(iT)] = R_X(mT), \quad (3.76)$$

where  $m = k - i$ , so the sampled process is WSS with autocorrelation equal to a sampled version of the autocorrelation  $R_X(\tau)$  of the original continuous-time signal. From (2.17), the power spectrum of the continuous-time random process and its sampled discrete-time process are related by

$$S_Y(e^{j\omega T}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} S_X(j(\omega - m\frac{2\pi}{T})). \quad (3.77)$$

As in the deterministic case, aliasing distortion results when the bandwidth is greater than half the sampling rate, where bandwidth in this case is defined in terms of the power spectrum.

**Example 3-10.**

Consider the approximation to thermal noise in Example 3-9. We wish to determine whether samples of such noise are uncorrelated; if they are, then sampled thermal noise is a discrete-time white Gaussian process. From (3.69), the autocorrelation function of the bandlimited noise  $W(t)$  is

$$R_W(\tau) = N_0 \frac{B}{\pi} \frac{\sin(B\tau)}{B\tau} \quad (3.78)$$

where  $B = 2\pi \times 10^{12}$ , or 1,000 GHz.  $R_W(\tau)$  has zero crossings at multiples of  $\pi/B$ , implying that samples of the random process taken at multiples of  $\pi/B$  will be uncorrelated. That is,

$$R_W(m\frac{\pi}{B}) = E[W(m\frac{\pi}{B})W(0)] = N_0 \frac{B}{\pi} \delta_m. \quad (3.79)$$

For these particular sampling rates, therefore, samples of the approximation to thermal noise are a discrete-time Gaussian white noise process. In practice, we are unlikely to sample any signal anywhere near the rate  $B/\pi$ , or 2,000 GHz. Since  $|R_W(\tau)|$  decays as  $\tau$  increases, samples at any reasonable sampling rate are *nearly* uncorrelated.  $\square$

Using the techniques discussed so far, we should have no difficulty considering systems that mix discrete and continuous-time random processes as well as deterministic signals. However, there are some subtleties. Consider a discrete-time random process  $X_k$  filtered by a continuous-time filter with impulse response  $h(t)$  in the sense defined in Section 2.1. The output can be written

$$Y(t) = \sum_{m=-\infty}^{\infty} X_m h(t-mT). \quad (3.80)$$

This is *pulse amplitude modulation* (PAM), described in detail in Chapter 6.

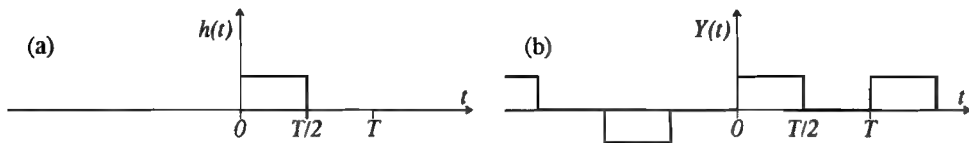
### Example 3-11.

The transmission of a discrete-time sequence of data symbols  $X_m$  over a continuous-time channel often takes the form of the random process in (3.80). Suppose that  $h(t)$  is as shown in Figure 3-5a and that  $X_k$  is a random sequence with i.i.d. samples taking values  $\pm 1$  with equal probability. A possible outcome is shown in Figure 3-5b. The first important observation is that the process  $Y(t)$  is not wide-sense stationary because  $E[Y(t+\tau)Y(t)]$  is not independent of  $t$ . For example,

$$E[Y(T/4)Y(0)] = E[X_0^2] = 1 \neq E[Y(T)Y(3T/4)] = 0.$$

This process is actually *cyclostationary*, a weaker form of stationarity. Since this process is not wide-sense stationary, its power spectrum is not defined.  $\square$

The fact that  $Y(t)$  in (3.80) is not wide-sense stationary is a major inconvenience. A common gimmick changes our random process into a wide-sense stationary process. Define the random variable  $\Theta$ , called a *random phase epoch*, that is uniformly



**Figure 3-5.** a. An example of a pulse shape for transmitting bits. b. An example of a waveform using this pulse shape.

distributed on  $[0, T]$  and independent of  $\{X_k\}$ . Then define the new random process

$$Z(t) = Y(t + \Theta) = \sum_{m=-\infty}^{\infty} X_m h(t + \Theta - mT). \quad (3.81)$$

This process has a *random phase* which is constant over time but chosen randomly at the beginning of time. Physically, this new process reflects our uncertainty about the phase of the signal; the origin in the time axis is of course arbitrary. This redefined process is wide-sense stationary, as shown in Appendix 3-A, with power spectrum

$$S_Z(j\omega) = \frac{1}{T} |H(j\omega)|^2 S_X(e^{j\omega T}). \quad (3.82)$$

Note the dependence on the power spectrum of the discrete-time process and the magnitude-squared spectrum of the pulse  $h(t)$ .

### Example 3-12.

Consider transmission of a random sequence of uncorrelated random variables  $X_k$  with equally probable values  $\pm 1$  using a pulse shape  $h(t)$ . The sequence  $X_k$  is white and the variance is unity, so the power spectrum of the data sequence is

$$S_X(e^{j\omega T}) = 1, \quad (3.83)$$

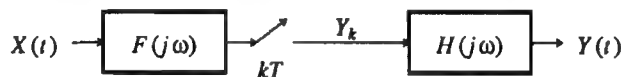
and the power spectrum of the random phase transmitted signal is

$$S_Z(j\omega) = \frac{1}{T} |H(j\omega)|^2. \quad (3.84)$$

With a white data sequence, the power spectrum has the shape of the magnitude squared of the Fourier transform of the pulse.  $\square$

## 3.2.6. Reconstruction of Sampled Signal

It might appear that (3.77) establishes the conditions under which a random process can be recovered from its samples, just as (2.17) does for deterministic signals. However, this appearance is deceiving because two random processes can have the same power spectrum and not be "equal" in any sense. The power spectrum is merely a second-order statistic, not a full characterization of the process. By a derivation similar to that in Appendix 3-A, we can investigate the recovery of the original continuous-time random process from its samples. A method of sampling and recovering a random process analogous to the deterministic case is shown in Figure 3-6. We first filter the random process using an anti-aliasing filter  $F(j\omega)$ , then sample, and



**Figure 3-6.** Sampling and recovery of a random process using anti-aliasing filter  $F(j\omega)$  and recovery filter  $H(j\omega)$ .

finally recover using recovery filter  $H(j\omega)$  to yield the random process  $Y(t)$ . To make  $Y(t)$  WSS we must again introduce a random phase. The way to tell whether the system recovers the input random process is not to calculate the output power spectrum, but rather to investigate the error signal between input and output. In particular, define

$$E(t) = X(t + \Theta) - Y(t + \Theta), \quad (3.85)$$

where  $\Theta$  is uniformly distributed over  $[0, T]$ . We would conclude that the recovery is exact (in a mean-square sense) if

$$E[|E(t)|^2] = 0. \quad (3.86)$$

This is not the same as showing that  $E(t) = 0$ , which cannot be shown using second order statistics only. However, (3.86) is just as good for engineering purposes. The conditions under which (3.86) is valid can be inferred from the following exercise, which can be solved using similar techniques to those used in Appendix 3-A.

#### Exercise 3-15.

Show that the power spectrum of  $E(t)$  is

$$\begin{aligned} S_E(j\omega) = & \frac{1}{T^2} |H(j\omega)|^2 \sum_{m \neq 0} S_X(j(\omega + m\frac{2\pi}{T})) |F(j(\omega + m\frac{2\pi}{T}))|^2 \\ & + |1 - \frac{1}{T} H(j\omega)F(j\omega)|^2 S_X(j\omega). \end{aligned} \quad (3.87)$$

□

Examining (3.87), the first term is aliasing distortion resulting from a signal at the output of the anti-aliasing filter, if it is not sufficiently bandlimited. In particular, if  $H(j\omega) = 0$  and  $F(j\omega) = 0$  for  $|\omega| \geq \pi/T$  then this term is identically zero. The second term is in-band distortion due to an improper reconstruction filter  $H(j\omega)$  and also distortion due to bandlimiting of the input prior to sampling. For an ideal reconstruction filter,

$$H(j\omega)F(j\omega) = T, \quad |\omega| < \pi/T, \quad (3.88)$$

in which case the error signal has power spectrum

$$S_E(j\omega) = \begin{cases} 0; & |\omega| < \pi/T \\ S_X(j\omega); & |\omega| \geq \pi/T \end{cases} \quad (3.89)$$

and the total error power is

$$E[E^2(t)] = 2 \cdot \frac{1}{2\pi} \int_{\pi/T}^{\infty} S_X(j\omega) d\omega. \quad (3.90)$$

The fact that the reconstruction error is just the error in initially bandlimiting  $X(t)$  is not surprising, and corresponds to the deterministic signal case.

The results of this subsection are important not only for their implications to the recovery of sampled random processes, but also in the techniques used. We will find

the need for similar techniques in the optimization problems of Chapter 9.

### 3.3. MARKOV CHAINS

A *discrete-time Markov process*  $\{\Psi_k\}$  is a random process that satisfies

$$p(\Psi_{k+1} | \Psi_k, \Psi_{k-1}, \dots) = p(\Psi_{k+1} | \Psi_k). \quad (3.91)$$

In words, the future sample  $\Psi_{k+1}$  is independent of past samples  $\Psi_{k-1}, \Psi_{k-2}, \dots$  if the present sample  $\Psi_k = \psi_k$  is known. The particular case of a Markov process where the samples take on values from a discrete and countable set  $\Omega_\Psi$  is called a *Markov chain*. In this section, we will often take  $\Omega_\Psi$  to be a set of integers. Markov chains are a useful model of a finite state machine with a random input, where the samples of the random input are statistically independent of one another. Since any digital circuit with internal memory (flip flops, registers, or RAMs) is a finite state machine, most digital communication systems contain finite state machines. Markov chains are useful signal generation models for digital communication systems with intersymbol interference or convolutional coding (Chapters 9, 13, and 14). Markov chain theory is also useful in the analysis of error propagation in decision-feedback equalizers (Chapter 10) and in the calculation of the power spectrum of line codes (Chapter 12). The following treatment uses Z-transform techniques familiar to the readers of this book. Sections 3.3.2 through 3.3.4, as well as Appendix 3-B, can be skipped on a first reading, since the techniques are not used until Chapter 10.

#### 3.3.1. State Transition Diagrams

Consider a random process  $\Psi_k$  (real, complex, or vector valued) whose sample outcomes are members of a finite or countably infinite set  $\Omega_\Psi$  of values. The random process  $\Psi_k$  is a Markov chain if (3.91) is satisfied. The next sample  $\Psi_{k+1}$  of a Markov chain is independent of the past samples  $\Psi_{k-1}, \Psi_{k-2}, \dots$  given the present sample  $\Psi_k$ . Furthermore, *all* future samples of the Markov chain are independent of the past given knowledge of the present, as shown in the following exercise.

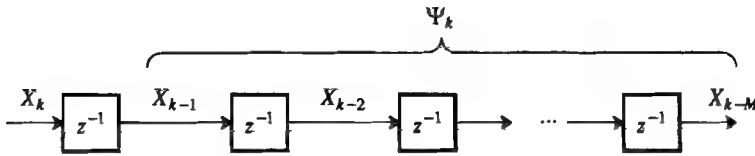
##### Exercise 3-16.

Let  $\Psi_k$  be a Markov chain and show that for any  $n > 0$ ,

$$p(\Psi_{k+n} | \Psi_k, \Psi_{k-1}, \dots) = p(\Psi_{k+n} | \Psi_k). \quad (3.92)$$

□

Since knowledge of the current sample  $\Psi_k$  makes the past samples irrelevant,  $\Psi_k$  is all we need to predict the future behavior of the Markov chain. For this reason,  $\Psi_k$  is said to be the *state* of the Markov chain at time  $k$ , and  $\Omega_\Psi$  is the set of all possible states.



**Figure 3-7.** A shift register process with independent inputs  $X_k$  is a Markov chain with state  $\Psi_k$ .

**Example 3-13.**

A *shift register process* is shown in Figure 3-7. If  $X_k$  is independent of  $X_{k-M-1}$ ,  $X_{k-M-2}$ ,  $\dots$  and we define the vector

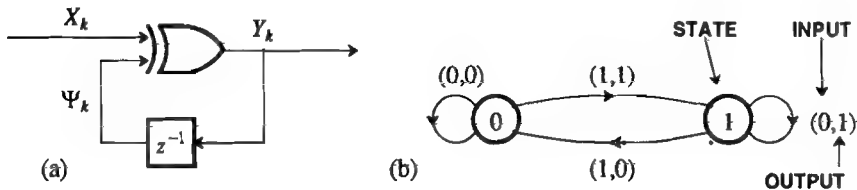
$$\Psi_k = [X_{k-1}, \dots, X_{k-M}] \quad (3.93)$$

( $M$  is the memory of the system) then the Markov property (3.91) is satisfied. This follows since  $\Psi_{k+1}$  is a function of  $X_{k+1}$  and  $\Psi_k$  only. Hence  $\{\Psi_k\}$  is a vector-valued discrete-time Markov process. If the inputs  $X_k$  are discrete-valued, then it is also a Markov chain.  $\square$

A Markov chain can be described graphically by a *state transition diagram*. This graph displays each state of the Markov chain as a node, and also displays the input and output or some other relevant properties for the transitions between states.

**Example 3-14.**

The *parity* of a bit stream  $X_k$  is defined to be the accumulated modulo-two summation of the bits, and is computed by the circuit in Figure 3-8a. It is sufficient for the input bits to be independent for the random process  $\Psi_k = Y_{k-1}$  to be Markov. It is easily seen from the diagram that  $\Psi_{k+1}$  depends only on the current state  $\Psi_k$  and the current input  $X_k$ .  $\Psi_k$  has a finite sample space  $\Omega_\Psi = \{0, 1\}$ , so the parity checker can be represented by the state transition diagram Figure 3-8b, where the arcs are labeled with the input that stimulates the state transition and the output resulting from the transition. The arcs of such a state diagram can alternatively be labeled with the transition probabilities, if the transition probabilities are independent of time.  $\square$



**Figure 3-8.** a. A circuit that computes the parity of the bit stream  $X_k$ . b. The state transition diagram of the corresponding Markov chain.



A Markov chain  $\Psi_k$  is called *homogeneous* if the conditional probability  $p(\Psi_k | \Psi_{k-1})$  is not a function of  $k$ . Homogeneity is therefore a kind of stationarity or time invariance. A homogeneous Markov chain can be characterized by its *state transition probabilities*, which we write with the shorthand

$$p(j|i) = p_{\Psi_{k+1}|\Psi_k}(j|i) \quad (3.94)$$

for  $i \in \Omega_\Psi$  and  $j \in \Omega_\Psi$ .

**Example 3-15.**

If in the previous example the incoming bits are not only independent but also identically distributed, then the Markov chain is homogeneous. If furthermore the incoming bits are equally likely to be one and zero, then the state transition probabilities are all 0.5.  $\square$

It is often convenient to define a random process that is some real-valued function of the state trajectory of a Markov chain,

$$X_k = f(\Psi_k). \quad (3.95)$$

This is encountered in the modeling of line coding (Chapter 12). The transmitted power spectrum is an important property of the line code, and thus we need to calculate the power spectrum of (3.95). This problem is considered in Appendix 3-B.

### 3.3.2. Transient Response of a Markov Chain

For a homogeneous Markov chain, we can find a relation for the evolution of the state probabilities with time. Using (3.33) we write

$$p_{k+1}(j) = \sum_{i \in \Omega_\Psi} p(j|i)p_k(i) \quad (3.96)$$

for all  $j \in \Omega_\Psi$ , where we have defined a notation for the probability of being in state  $i$  at time  $k$ ,

$$p_k(i) = \Pr\{\Psi_k = i\}. \quad (3.97)$$

The new notation emphasizes that  $p_k(i)$  is a discrete-time sequence. In applications we often want to determine the probability of being in a certain state  $j$  at a certain time  $k$  given a set of probabilities for being in those states at initial time  $k=0$ . We can accomplish this by analyzing (3.96), a system of *time-invariant* difference equations, using Z-transform techniques. If we define  $p_k(j)=0$  for  $k < 0$ , then the Z-transform of the state probability for state  $j$  is

$$P_j(z) = \sum_{k=0}^{\infty} p_k(j)z^{-k}. \quad (3.98)$$

**Exercise 3-17.**

Take the Z-transform of both sides of (3.96) to show that

$$P_j(z) = p_0(j) + \sum_{i \in \Omega_\Psi} p(j|i)z^{-1}P_i(z). \quad (3.99)$$

$\square$

If there are  $N$  states, (3.99) gives us  $N$  equations with  $N$  unknowns  $P_j(z)$ . These equations can be solved and the inverse Z-transform calculated to determine the state probability  $p_k(i)$ .

### Example 3-16.

Continuing Example 3-14, the parity check circuit, suppose that the initial state is equally likely to be either zero or one, so

$$p_0(0) = p_0(1) = 0.5. \quad (3.100)$$

Suppose further that the incoming bits  $X_k$  are equally likely to be zero or one, so the transition probabilities  $p(j|i)$  are all  $1/2$ . Then (3.99) becomes

$$\begin{aligned} P_0(z) &= 0.5 + 0.5z^{-1}P_0(z) + 0.5z^{-1}P_1(z) \\ P_1(z) &= 0.5 + 0.5z^{-1}P_1(z) + 0.5z^{-1}P_0(z). \end{aligned} \quad (3.101)$$

Solving this set of two simultaneous equations, the Z-transforms of the state probabilities are equal,

$$P_0(z) = P_1(z) = \frac{0.5}{1 - z^{-1}}. \quad (3.102)$$

Using the Z-transform pair in Problem 2-15 we can invert the Z-transform to get

$$p_k(0) = p_k(1) = 0.5 \cdot u_k \quad (3.103)$$

where  $u_k$  is the unit step function. The chain is therefore equally likely to be in either state at any point in time beginning at  $k = 0$ . A Markov chain in which the state probabilities are independent of time is called *stationary*.  $\square$

### 3.3.3. Signal Flow Graph Analysis

Translation of a state diagram into a set of equations to be solved is often made easier using signal flow graphs. A signal flow graph is a graphical representation of a linear equation, and in particular can represent the system of equations given by (3.99). Its value lies in the fact that the state diagram can be directly translated into a topologically equivalent signal flow graph representing the equations. In fact, the experienced can write down the signal flow graph directly without ever generating a state diagram. The idea of a graph representing linear equations is illustrated by the following simple example.

### Example 3-17.

The equation  $w = au + x$  can be represented by the signal flow graph shown in Figure 3-9a. The nodes of the graph represent the variables  $u$ ,  $w$ , and  $x$ , while the two arcs represent the multiplication of the variables by constants, and also the addition. The signal flow graph in Figure 3-9b represents the recursive equation  $x = au + bw + cx$ .  $\square$

In general, a node in a signal flow graph represents a variable that is equal to the sum of the incoming arcs. A weight on an arc is a multiplicative factor. Our interest is in signal flow graphs in which the variables are all Z-transforms.

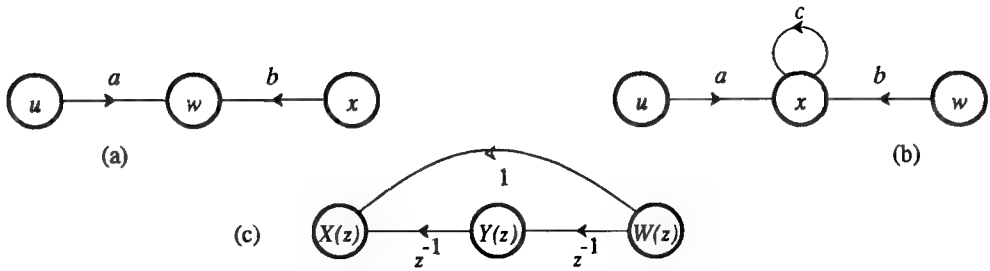


Figure 3-9. Several signal flow graphs representing linear equations.

### Example 3-18.

The signal flow graph in Figure 3-9c represents a dynamical system described by the equations  $X(z) = z^{-1}Y(z) + W(z)$  and  $Y(z) = z^{-1}W(z)$ .  $\square$

From the last example, it is clear that the equations (3.99) can be represented using a signal flow graph for any given Markov chain, as shown in Figure 3-10. Shown are just two of the states,  $i$  and  $j$ . Each of the states is represented by two nodes of the graph, one for the Z-transform of the state probability sequence,  $P_i(z)$ , and the other for the initial probability of that state  $p_0(i)$  (the latter is not a variable in the equations, but a constant). In many cases the initial probability is zero so the corresponding node can be omitted.

### Example 3-19.

Returning to the parity check example of Example 3-14, the equations (3.101) are represented by the signal flow graph in Figure 3-11. Note that this one figure takes the place of the state diagram of Figure 3-8 and the set of equations of (3.101).  $\square$

In retrospect, the signal flow graph is intuitive. Each state transition has a delay

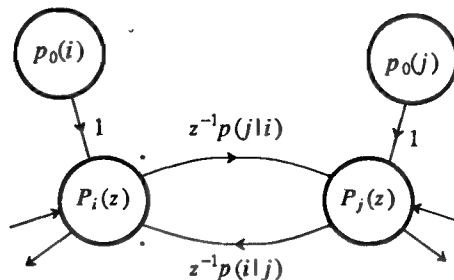
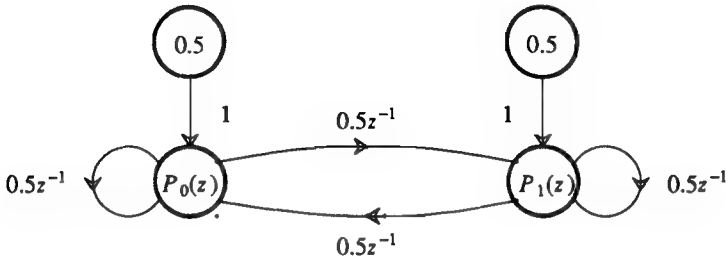


Figure 3-10. A signal flow graph representation of the Markov chain dynamical equations (3.99).



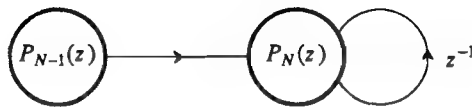
**Figure 3-11.** A signal flow graph representation of the system of state probabilities for the parity examples.

operator  $z^{-1}$  corresponding to the time it takes for that transition to occur, as well as the probability of that transition. The arcs from the initial state probabilities have no such delay since the initialization is instantaneous, and we can think of that transition as occurring only once at  $k = 0$ . For Markov chains that start in a particular state, there will only be one such node corresponding to the starting state.

Once we have a signal flow graph, we can easily write down the set of equations and then solve them for the  $Z$ -transform of the state probabilities. For some problems, a shortcut known as *Mason's gain formula* allows us to solve these equations directly by inspection of the signal flow graph [2,3,4,5].

### 3.3.4. First Passage Problem

When we use Markov chains to model the behavior of framing recovery circuits (Chapter 19) and error propagation (Chapter 10), we would like to calculate the *average first passage time* for an *absorption state* of the chain. An absorption state is defined as a state with an entry but no exit, so that the steady-state probability of that state is unity. This is illustrated in Figure 3-12 for the case where the absorption state is  $N$ . An absorption state must have a self-loop with gain  $z^{-1}$  indicating that the chain stays in that state forever. The figure also assumes that there is only one way to get to the absorption state, from state  $N-1$ , although that is not necessary for the following analysis.



**Figure 3-12.** A part of a signal flow graph for a Markov chain in which state  $N$  is an absorption state, with only one entry from the outside, namely from state  $N-1$ .

What we are often interested in is the *first passage time to state  $N$* , which is defined as the time-index of the first time we enter that state. Define the probability of entering state  $N$  at time  $k$  as  $q_k(N)$ . Then we have that

$$p_k(N) = p_{k-1}(N) + q_k(N), \quad (3.104)$$

or in words, the probability of being in state  $N$  at time  $k$  is equal to the probability of being in that state at time  $k-1$  plus the probability of first entry into that state at time  $k$ . This relation follows from the fact that there are only two mutually exclusive ways to be in state  $N$  at time  $k$  — either we were there before or else we entered the state at time  $k$ . From (3.104) we can relate the first passage probability to the state probability that has already been calculated. Assuming that  $p_0(N) = 0$ , taking the Z transform of (3.104) we get

$$Q_N(z) = (1 - z^{-1})P_N(z). \quad (3.105)$$

Since  $P_N(z)$  is an absorption state, it turns out that it will always have a factor of  $(1 - z^{-1})$  in the denominator which will be canceled, resulting in a  $Q_N(z)$  which is simpler than the  $P_N(z)$  that we started with.

If we define the average or expected time for first entry into state  $N$  as  $f_N$ , then it turns out that we can find this time without the need to take the inverse Z-transform of  $Q_N(z)$ .

### Exercise 3-18.

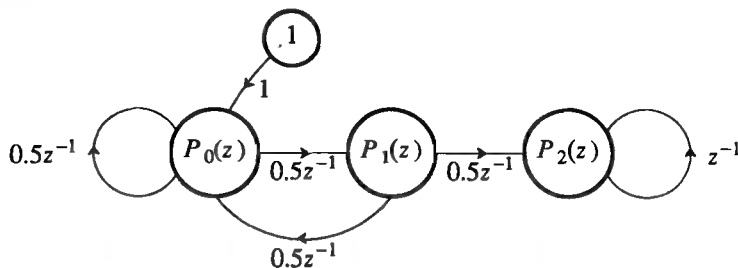
Show that the mean first passage time is

$$f_N = - \frac{\partial}{\partial z} Q_N(z) \Big|_{z=1}. \quad (3.106)$$

□

### Example 3-20.

If we toss a fair coin, what is the average number of tosses until we have seen two heads in a row? The signal flow graph for this example is shown below:



The numbering of states is the number of heads in a row. We assume that we start with zero heads in a row. At each toss the number of heads in a row increases by one with probability  $\frac{1}{2}$ , or goes back to zero with probability  $\frac{1}{2}$  (that is, we get a tail). We define state two (two heads in a row) as an absorption state so that we can calculate the first passage time. Solving the linear equations, we get

$$P_2(z) = \frac{z}{4z^3 - 6z^2 + z + 1} = \frac{z}{(z-1)(4z^2 - 2z - 1)}. \quad (3.107)$$

Finally,

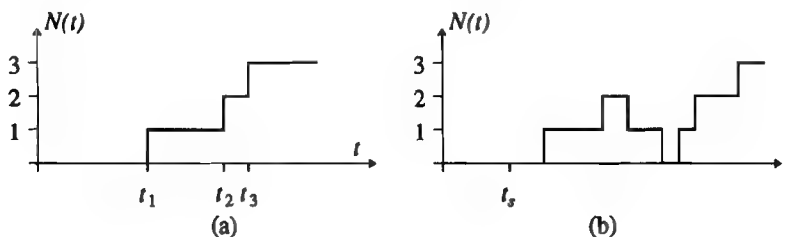
$$f_N = -\frac{\partial}{\partial z} \frac{1}{4z^4 - 2z - 1} \Big|_{z=1} = 6. \quad (3.108)$$

□

### 3.4. THE POISSON PROCESS AND QUEUEING

There was a time when no random processes could challenge the Gaussian process for the attention of communication theorists. However, the *Poisson process*, and its generalization, the *birth and death process* can reasonably claim to hold that distinction. The question often arises in communications as to the distribution for the times of discrete events, such as the arrivals of messages at a digital communication multiplex, or the arrivals of photons in a light beam at an optical detector in an optical communication system. The Poisson process models the most random such distribution, and is an excellent model for many of these situations.

To proceed, we need to define the notion of *random points in time*, where a point in time might denote the arrival of a message from a random source or a photon at a photodetector. Defining some notation, let the time of the  $k$ -th arrival be denoted by  $t_k$ , where of course  $t_k \geq t_j$  for  $k > j$ . Further, define a continuous-time random process  $N(t)$  that equals the number of arrivals from some starting time  $t_0$  to the current time  $t$ . We call  $N(t)$  a *counting process* since it counts the accumulated number of random points in time. Thus,  $N(t)$  assumes only non-negative integer values, has initial condition  $N(t_0) = 0$ , and at each random point in time  $t_k$ ,  $N(t)$  increases by one. Such a counting process is pictured in Figure 3-13a, where the arrival times and the value of the counting process are pictured for one typical outcome.



**Figure 3-13.** Typical outcomes from a counting process  $N(t)$ . a. A counting process which is monotone increasing. b. A counting process, which has both arrivals and departures and hence can increase or decrease.

In some situations there are only arrivals, so that a counting process of the type pictured in Figure 3-13a is the appropriate model. In other situations, there are departures as well as arrivals. A typical situation is the *queue* pictured in Figure 3-14. We can define a counting process  $N(t)$  to be the difference between the accumulated number of arrivals and the accumulated number of departures.

### Example 3-21.

Consider a computer communication system that stores arriving messages in a buffer before retransmitting them to some other location.  $N(t)$  gives a current count of the number of messages in the system at time  $t$ . A typical outcome of such a process is pictured in Figure 3-13b, where it should be noted that the process can never go below zero (since nothing can depart if there is nothing in the buffer).  $\square$

In many instances of practical importance, the count  $N(t)$  at time  $t$  is all we need to know to predict the future evolution of the system after time  $t$ . The manner in which system reached  $N(t)$  is irrelevant in terms of predicting the future. For this case, the counting process denotes the *state of the system* in the same sense as Markov chains in the last section. In particular, we say that the system is in state  $j$  at time  $t$  if  $N(t) = j$ . This is similar to a Markov chain with one important distinction — a Markov chain can only change states at discrete points in time, whereas we now allow the state to change at any continuous point in time. Like Markov chains, a sample of the counting process  $N(t_0)$  is a discrete-valued random variable. Just as for Markov chains (3.97), we define a probability of being in state  $j$  at time  $t$  as

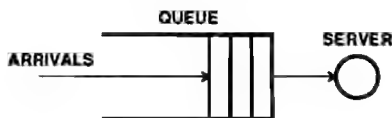
$$q_j(t) = \Pr[N(t) = j] = P_{N(t)}(j). \quad (3.109)$$

This notation emphasizes that this probability is a continuous-time function. The only real distinction between (3.97) and (3.109) is that the later is defined for continuous-time and the former for discrete-time.

In the following subsections, we analyze a counting process under the specific conditions appropriate for optical communication (Section 3.4.3) and statistical multiplexing (Section 3.4.2).

### 3.4.1. Birth and Death Process

The cases of interest to us are subsumed by a general process called a *birth and death process*, which is a mathematician's macabre terminology for a counting process with both arrivals and departures. This analysis is given in this section.



**Figure 3-14.** A queueing system, which models among other things the status of a buffer in a communication system.

We have to somehow model the evolution of the system from one state to another. The approach for the Markov chain in (3.94) is inappropriate, since the probability of transition between any two states at any point in time  $t$  is most likely zero! While we cannot characterize the *probability* of transition, what we can characterize is the *rate* of transition between two states. Suppose for two particular states, the rate of transitions between one state and the other is a constant  $R$ . What we mean by this is that in a time  $\delta t$  we can expect an average  $R \delta t$  transitions. If  $\delta t$  is very small, then  $R \delta t$  is a number much smaller than unity, and the probability of more than one transition in time  $\delta t$  is vanishingly small. Under these conditions, we can think of  $R \delta t$  as the probability of one transition in time  $\delta t$ , and  $(1 - R \delta t)$  as the probability of no transition.

This logic leads us to a transition diagram and associated set of differential equations. The transition diagram in Figure 3-15 associates a node with each state, and within that node we put the probability of being in that state at time  $t$ , which we denote  $q_j(t)$ . Each transition in the diagram is labeled with the rate at which that transition occurs, where the rates in the general case are allowed to be time-varying (non-homogeneous). Each rate is labeled with a subscript indicating the state in which it originates, where  $\lambda(t)$  is the rate for transitions corresponding to births or arrivals and  $\mu(t)$  corresponds to deaths or departures. Reiterating, the interpretation of these rates is as follows: for a very small time interval  $\delta t$ , the probability of a particular transition is equal to the rate times the time interval.

The set of differential equations which describe the evolution of the birth and death process are

$$\begin{aligned} \frac{dq_j(t)}{dt} &= \lambda_{j-1}(t)q_{j-1}(t) + \mu_{j+1}(t)q_{j+1}(t) - (\lambda_j(t) + \mu_j(t))q_j(t), \quad j \geq 0 \\ q_{-1}(t) &= 0. \end{aligned} \quad (3.110)$$

These equations can be derived rigorously from fundamental principles [6], but for our purposes they are evident from intuitive considerations. The equations say that the rate of increase of a probability with time for state  $j$  is equal to the rate at which transitions into that state from states  $j-1$  and  $j+1$  are occurring (times the current probability of those states) minus the rate at which transitions out of state  $j$  are occurring (times the current probability of state  $j$ ).

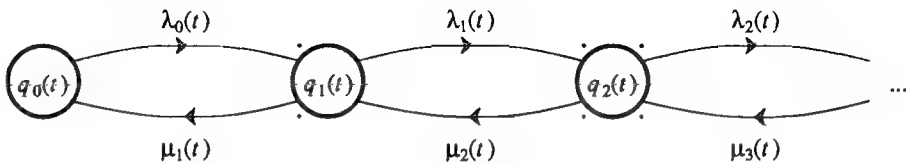


Figure 3-15. State transition diagram for a birth and death process.



We must also specify an initial condition, which for our purposes specifies that the process starts in state zero (no arrivals) at time  $t_0$ ,

$$q_j(t_0) = \begin{cases} 1, & j = 0 \\ 0, & j > 0 \end{cases} \quad (3.111)$$

The first order differential equations can be solved for many special cases.

### Example 3-22.

Consider the important case of a *pure birth process* in which  $\mu_j(t) = 0$ . Also assume the birth rates are all the same and a constant with time,  $\lambda_j(t) = \lambda$ . The transition diagram for this model is shown in Figure 3-16. This corresponds to the important case where the arrival rate does not depend on the state of the system, the usual case in the problems that we will encounter. Then (3.110) becomes

$$\frac{dq_j(t)}{dt} + \lambda q_j(t) = \lambda q_{j-1}(t) \quad (3.112)$$

which is a simple first order differential equation with constant coefficients. Assume that the initial condition is

$$q_0(0) = 1 \quad (3.113)$$

implying that the initial count at  $t = 0$  is 0. We can solve this using very similar techniques to our solution of the Markov chain, but use the Laplace transform in place of the Z-transform. In analogy to (3.98), defining the Laplace transform of the state probability,

$$Q_j(s) = \int_0^{\infty} q_j(t) e^{-st} dt \quad (3.114)$$

Taking the Laplace transform of both sides of (3.112),

$$sQ_j(s) - q_j(0) + \lambda Q_j(s) = \lambda Q_{j-1}(s) \quad (3.115)$$

Using (3.111), with  $t_0 = 0$ , this becomes

$$Q_0(s) = \frac{1}{s + \lambda}, \quad Q_j(s) = \frac{\lambda}{s + \lambda} Q_{j-1}(s), \quad j > 0 \quad (3.116)$$

This set of iterative equations for the state probability Laplace transform is easily solved by iteration,

$$Q_j(s) = \frac{\lambda^j}{(s + \lambda)^{j+1}} \quad (3.117)$$

and taking the inverse Laplace transform, we find that for  $t \geq 0$  and  $j \geq 0$ ,

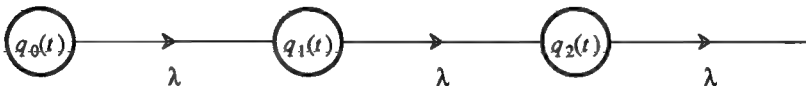


Figure 3-16. State transition diagram for a constant-rate pure birth process.

$$\Pr[N(t) = j] = q_j(t) = \frac{(\lambda t)^j}{j!} e^{-\lambda t}. \quad (3.118)$$

This is the well-known *Poisson distribution* with parameter  $\lambda t$ . For this reason, the pure birth process  $N(t)$  we have just analyzed is called a *Poisson process* with constant rate. We will generalize this to a variable rate in the next subsection.  $\square$

The Poisson distribution is an important one in the theory of birth and death processes, so we summarize its properties in the following exercise.

#### Exercise 3-19.

Consider a Poisson distribution with parameter  $a$ ,

$$p_N(k) = e^{-a} \frac{a^k}{k!}. \quad (3.119)$$

- (a) Show that the mean and variance of this distribution are

$$E[N] = a, \quad \text{Var}[N] = a. \quad (3.120)$$

(Hint: Form a power series for  $e^a$  and differentiate it twice.)

- (b) Show that the moment generating function is given by

$$\log_e \Phi_N(s) = a(e^s - 1). \quad (3.121)$$

$\square$

The last example can be generalized with respect to the initial condition.

#### Exercise 3-20.

Show that if  $N(t_0) = k$  (there have been  $k$  counts up to time  $t_0$ ), then

$$q_j(t) = \frac{(\lambda(t - t_0))^{j-k}}{(j-k)!} e^{-\lambda(t - t_0)}, \quad j \geq k, t \geq t_0. \quad (3.122)$$

$\square$

This result implies that the number of counts starting at  $t = t_0$  is a Poisson distribution with parameter  $\lambda(t - t_0)$ , which is the expected number of arrivals since the start time. Furthermore, index  $j-k$  of the Poisson distribution is the number of counts since the start time. The important conclusion is that the number of arrivals in the interval starting at  $t = t_0$  has a distribution which does not depend in any way on what happened prior to  $t_0$ . This is roughly the definition of a *Markov process*, and a Poisson counting process is in fact a Markov process. For such a process, the number of arrivals in the interval  $[t_0, t]$  is statistically independent of the number of arrivals in any other nonoverlapping interval of time. It is in this sense that the Poisson process is the most random among all monotone non-decreasing counting processes.

#### Exercise 3-21.

(Pure death process.) For  $\lambda_j(t) = 0$ , consider the case where departures from the system are proportional to the state index,  $\mu_j(t) = j\mu$ . This is an appropriate model for a system in which the departure or death rate is proportional to the size of the population, as in a

human population. Further, assume that the initial state at  $t = 0$  is  $n$ . Draw the state transition diagram and show that the state probabilities obey a binomial distribution,

$$\Pr[N(t) = j] = q_j(t) = \binom{n}{j} p^j(t) (1 - p(t))^{n-j}, \quad p(t) = e^{-\mu t}. \quad (3.123)$$

□

Now we give an example of a problem in which both births and deaths occur. This is an example of a *queueing problem*, and it is appropriate at this point to define some terminology used in queueing, particularly as it relates to digital communication. A queue is a *buffer* or *memory* which stores messages. There is some mechanism which clears messages from the queue, which is usually the transmission of the message to another location. This mechanism is called the *server* to the queue. Assume a server can process only one message at a time, so that if more than one message is being processed (there are multiple communication channels for transmission of messages), then there are an equivalent number of servers. Typically the buffer contains space for a maximum number of messages to wait for service, and the number of messages that can be waiting at any time is called the *number of waiting positions*. The *state of the system*, which naturally tracks a counting process, is the number of messages waiting for service plus the number of messages currently being served. Messages arrive at the queue (births) at random times, and they depart from the queue (deaths) due to the completion of service.

### Exercise 3-22.

(Queue with one server and no waiting positions.) Assume that a queue has constant arrival rate  $\lambda$ , a single server which clears a message being served at rate  $\mu$ , and no waiting positions. If a message arrives while the server is busy then since there are no waiting positions that arrival is lost and leaves the system permanently. Draw the state transition diagram for the system and show that the probability that the server is not busy is

$$q_0(t) = \frac{\mu}{\lambda + \mu} + \left[ q_0(0) - \frac{\mu}{\mu + \lambda} \right] e^{-(\mu + \lambda)t}. \quad (3.124)$$

□

The differential equation approach we have described is capable of describing the transient response of a system starting with any initial condition. Often, however, it is sufficient to know what the state probabilities are in the steady state. There is no such steady state distribution for a Poisson process, since the state grows without bound. However, for queueing systems where the service rate is always guaranteed to be higher than the arrival rate, and where all the rates are independent of time, there will be a steady state distribution. This distribution can be obtained by letting  $t \rightarrow \infty$  in the transient solution we have obtained, or can be obtained much more simply by setting the time derivatives in the differential equations to zero and solving for the resulting probabilities.

**Example 3-23.**

Continuing Exercise 3-22, letting  $t \rightarrow \infty$  in (3.124), the steady state probability is

$$q_0(\infty) = \frac{\mu}{\lambda + \mu}. \quad (3.125)$$

We can get this same result without solving the differential equation by setting the derivative in (3.110) to zero.  $\square$

In the following two sections we will specialize the general birth and death process to two situations of particular interest to us.

**3.4.2. M/M/1 Queue**

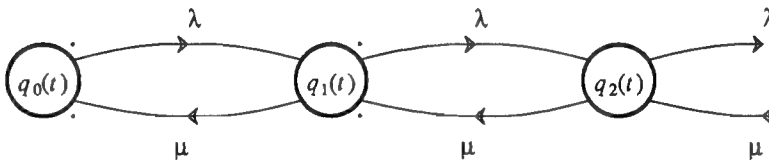
Consider the following queueing model which characterizes a single server queue with the most mathematically tractable assumptions. This model is actually a combination of the pure birth process of Example 3-22 and a pure death process (Exercise 3-21). Assume arrivals occur at a constant rate  $\lambda$  independent of the number of waiting positions occupied, there are an infinite number of waiting positions so that no arrival ever encounters a full buffer, arrivals wait indefinitely for service, and there is a single server with service rate  $\mu$ . The departure rate is independent of the number of messages waiting in the queue, as long as there is at least one. The state transition diagram for this queueing model is shown in Figure 3-17.

As in most queueing problems, we are content to know the steady state distribution of states. This distribution will only exist if the service rate  $\mu$  is greater than the arrival rate  $\lambda$ , because otherwise the buffer size will grow to infinity. Making that assumption, the differential equations governing the queue are

$$\begin{aligned} \frac{dq_j(t)}{dt} &= \lambda q_{j-1}(t) + \mu q_{j+1}(t) - (\lambda + \mu)q_j(t), \quad j > 0 \\ \frac{dq_0(t)}{dt} &= \mu q_1(t) - \lambda q_0(t) \end{aligned} \quad (3.126)$$

with initial condition (assuming there are no positions occupied at time  $t = 0$ ),

$$q_j(0) = \begin{cases} 1, & j = 0 \\ 0, & j > 0 \end{cases}. \quad (3.127)$$



**Figure 3-17.** The state transition diagram for the single server queue with an infinite number of waiting positions.

We could attempt to solve this system of differential equations, but since we are content with the steady state solution, set the derivatives to zero,

$$\begin{aligned} 0 &= \lambda q_{j-1} + \mu q_{j+1} - (\lambda + \mu) q_j; \quad j > 0 \\ 0 &= \mu q_1 - \lambda q_0 \end{aligned} \quad (3.128)$$

where we have also taken the liberty of suppressing the time dependence since we are looking only at the steady state. These equations are easily solved.

### Exercise 3-23.

Show that the solution to (3.128) is

$$q_j = \rho^j (1 - \rho) \quad (3.129)$$

where  $\rho$  is called the *offered load*,

$$\rho = \lambda / \mu \quad (3.130)$$

and is less than unity by assumption. Note from (3.129) that the probability that the single server is busy is  $1 - q_0 = \rho$ , which is obvious since the server has more "capacity" than the arrivals require by a factor of  $\mu/\lambda$ . Thus,  $\rho$  is also called the *server utilization*.  $\square$

In many queueing problems the most critical parameter is the delay that a new arrival experiences before being served. This is also called the *queueing delay*, and represents a significant impairment in communication systems that utilize a buffer delay discipline to increase the capacity of a communication link (Chapter 18). A related parameter is the *waiting time*, which is defined to be the queueing delay plus the service time. The calculation of the delay is a little more complicated than what we have done heretofore, so we will simply state the results [6]. The mean delay is given by

$$D = \frac{\rho}{\mu(1 - \rho)}. \quad (3.131)$$

Note that as the offered load or server utilization approaches unity, the mean delay grows without bound; conversely, as the utilization approaches zero, the lightly loaded queue, the delay approaches zero. The mean queueing delay is equal to the average service time  $1/\mu$  for a utilization of  $\rho = 1/2$ .

### 3.4.3. Poisson Process With Time-Varying Rate

In optical communication systems, the counting process which gives the accumulated number of arrival times for photons is a Poisson process (Section 5.3). The Poisson process is a pure birth process where the arrival rate is independent of the state of the system, and we have already been exposed to it in Example 3-22 for a constant arrival rate. In optical communication, the arrival rate is actually signal dependent, so in this section we discuss that case.

The Poisson process with time-varying rate is the pure birth process in which the incoming rate  $\lambda(t)$  is independent of the state of the system. Thus, the system is governed by a first-order differential equation with time-varying coefficients,

$$\frac{dq_j(t)}{dt} + \lambda(t)q_j(t) = \lambda(t)q_{j-1}(t), \quad q_{-1}(t) = 0, \quad (3.132)$$

and we assume the system starts at time  $t_0$  in state  $j = 0$ . Because of the time-varying coefficients, the Laplace transform is of no help, and we must resort to solving the differential equation directly. This is straightforward (since it is a first order equation), but tedious, so the solution is relegated to Appendix 3-C. Define

$$\Lambda(t) = \int_{t_0}^t \lambda(u) du, \quad (3.133)$$

which has the interpretation as the average total number of arrivals in the interval  $[t_0, t]$ . Then the probability of  $n$  arrivals in the interval  $[t_0, t]$  is governed by a Poisson distribution with parameter  $\Lambda(t)$ ,

$$q_n(t) = \frac{\Lambda^n(t)}{n!} e^{-\Lambda(t)}. \quad (3.134)$$

This reduces to the solution given in Example 3-22 for the constant rate case.

As a reminder, (3.134) specifies the number  $N(t)$  of arrivals during the time interval  $[t_0, t]$ . This random number of arrivals is Poisson distributed with parameter  $\Lambda(t)$ , and hence has mean and variance

$$E[N(t)] = \Lambda(t), \quad \sigma_{N(t)}^2 = \Lambda(t). \quad (3.135)$$

As in the constant rate case, it can be shown that the number of arrivals in any two non-overlapping intervals are statistically independent.

### 3.4.4. Shot Noise

In optical communication, a waveform is generated in the photodetector by generating impulses at times corresponding to random arrival times of photons and then filtering these impulses. This is known as a *filtered Poisson process*, or a *shot noise process*.

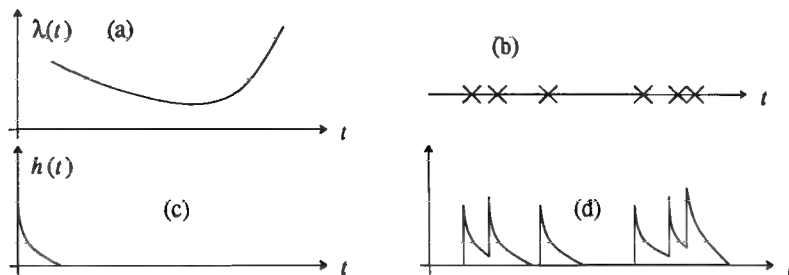
If a Poisson process is characterized by a set of arrival times  $t_k$  for the  $k$ -th arrival, and given a filter with impulse response  $h(t)$ , then a shot noise process is a continuous-time random process  $X(t)$  with outcome

$$x(t) = \sum_k h(t - t_k). \quad (3.136)$$

An outcome of this random process is illustrated in Figure 3-18 for a particular impulse response. In this figure, it is assumed that qualitatively the duration of the impulse response is short relative to the average time between arrivals. If the impulse response were long, this would have an averaging effect resulting in a much smoother outcome.

It is shown in Appendix 3-D that the moment generating function of the shot noise process at time  $t$  is

$$\log_e \Phi_{X(t)}(s) = \lambda(t) * (e^{sh(t)} - 1). \quad (3.137)$$



**Figure 3-18.** Illustration of an outcome of a shot noise process. a. The average arrival rate vs. time. b. The random actual times of arrival, where arrivals occur at the average rate given in a. c. The impulse response of the filter. d. The corresponding outcome.

The mean and variance of shot noise are easily derived from (3.137).

#### Exercise 3-24.

Show that the mean value of shot noise is the convolution of the filter impulse response with the arrival rate,

$$m_X(t) = E[X(t)] = \lambda(t) * h(t) \quad (3.138)$$

and that the variance is the convolution of the square of the filter impulse response with the arrival rate,

$$\sigma_X^2(t) = E[X^2(t)] - m_X^2(t) = \lambda(t) * h^2(t). \quad (3.139)$$

These relations are known as *Campbell's theorem*.  $\square$

### 3.4.5. High-Intensity Shot Noise

When the intensity of shot noise is high, the statistics become that of a Gaussian random process. The intuition behind this is that  $X(t)$  is the sum of a large number of independent events, and hence approaches a Gaussian by the central limit theorem. To demonstrate this more rigorously, we will show that the moment generating function of shot noise approaches a Gaussian moment generating function in the limit of high intensity.

In order to avoid an infinitely large power of shot noise, as the intensity grows we need to scale the size of the impulse response  $h(t)$  also. Therefore, let us use a scaling constant  $\beta$ , which we will allow to grow to infinity, and let

$$\lambda(t) = \beta \lambda_0(t), \quad h(t) = \frac{1}{\sqrt{\beta}} h_0(t). \quad (3.140)$$

With this scaling, we get from Campbell's theorem that

$$m_X(t) = \sqrt{\beta} \lambda_0(t) * h_0(t), \quad \sigma_X^2(t) = \lambda_0(t) * h_0^2(t). \quad (3.141)$$

Hence, as the scaling factor  $\beta$  grows, the variance of the process stays constant and the mean value grows without bound. We cannot help this, because as the intensity grows

the variance becomes a smaller fraction of the mean. In this sense high-intensity shot noise approaches a deterministic signal  $m_X(t)$  as the intensity grows.

Only two terms in the moment generating function are important as the scaling constant  $\beta$  grows.

**Exercise 3-25.**

Show that for large  $\beta$  the only significant terms in the moment generating function of (3.137) are

$$\log_e \Phi_{X(t)}(s) \approx s \sqrt{\beta} \lambda_0(t) * h_0(t) + 0.5 s^2 \lambda_0(t) * h_0^2(t) \quad (3.142)$$

Comparing this with the Gaussian moment generating function of (3.41), we see that high intensity shot noise is approximately Gaussian with mean and variance given by (3.141).  $\square$

### 3.4.6. Random-Multiplier Shot Noise.

In optical communication systems, it is sometimes appropriate to introduce a random multiplier into the shot noise process, *viz.*

$$X(t) = \sum_k G_k h(t-t_k) \quad (3.143)$$

where  $G_k$  is a sequence of mutually statistically independent identically distributed random variables which are also statistically independent of the arrival times  $t_j$  for all  $j$ .

**Exercise 3-26.**

Use Campbell's theorem and the assumptions to show that the mean-value of (3.143) is

$$m_X(t) = E[G] \lambda(t) * h(t) \quad (3.144)$$

and the variance is

$$\sigma_X^2(t) = E[G^2] \lambda(t) * h^2(t) \quad (3.145)$$

where  $E[G]$  and  $E[G^2]$  are the mean-value and second moment of the random multiplier  $G_k$  for all  $k$ .  $\square$

## 3.5. FURTHER READING

For a general introduction to random variables and processes, Papoulis [7], Stark and Woods [8], and Ross [9] are recommended. Papoulis has more of an engineering perspective. Both books have comprehensive treatments of Markov chains and Poisson and shot noise processes. An excellent introduction to Poisson processes can be found in Ross [10]. There are a number of books that give comprehensive treatment to the application of Poisson and birth and death processes to queueing models, such as Cooper [6], Hayes [11], and Kleinrock [12].



### APPENDIX 3-A POWER SPECTRUM OF A CYCLOSTATIONARY PROCESS

In this appendix we determine the power spectrum of the PAM random process with a random phase epoch (3.81). Calculating the autocorrelation function of (3.81),

$$\begin{aligned} E[Z(t+\tau)Z^*(t)] &= E[Y(t+\Theta+\tau)Y^*(t+\Theta)] \\ &= E\left[\sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} X_m X_n^* h(t+\Theta-mT+\tau)h^*(t+\Theta-nT)\right]. \end{aligned} \quad (3.146)$$

Assuming we can interchange expectation and summation, we use the fact that  $\Theta$  is independent of  $X_k$  to get

$$E[Z(t+\tau)Z^*(t)] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} E[X_m X_n^*] E[h(t+\Theta-mT+\tau)h^*(t+\Theta-nT)]. \quad (3.147)$$

The first expected value is simply the autocorrelation function  $R_X(m-n)$ . The second expected value can be computed using the definition of expectation and the p.d.f. of the uniform random variable  $\Theta$

$$E[h(t+\Theta-mT+\tau)h^*(t+\Theta-nT)] = \int_0^T \frac{1}{T} h(t+\theta-mT+\tau)h^*(t+\theta-nT) d\theta. \quad (3.148)$$

Changing variables, letting  $i = m-n$ , using (3.148), and exchanging summations, we get

$$E[Z(t+\tau)Z^*(t)] = \frac{1}{T} \sum_{i=-\infty}^{\infty} R_X(i) \sum_{n=-\infty}^{\infty} \int_0^T h(t+\theta-(i+n)T+\tau)h^*(t+\theta-nT) d\theta. \quad (3.149)$$

Changing variables again and defining  $\alpha = t+\theta-nT$ , we get

$$E[Z(t+\tau)Z^*(t)] = \frac{1}{T} \sum_{i=-\infty}^{\infty} R_X(i) \sum_{n=-\infty}^{\infty} \int_{t-nT}^{t-nT+T} h(\alpha-iT+\tau)h^*(\alpha) d\alpha. \quad (3.150)$$

The second summation is the sum of integrals with adjoining limits, so it can be replaced with a single infinite integral

$$E[Z(t+\tau)Z^*(t)] = \frac{1}{T} \sum_{i=-\infty}^{\infty} R_X(i) \int_{-\infty}^{\infty} h(\alpha-iT+\tau)h^*(\alpha) d\alpha, \quad (3.151)$$

which is independent of  $t$ , so the process  $Z(t)$  is wide sense stationary. To get the power spectrum, we take the Fourier transform with  $\tau$  as the time index

$$S_Z(j\omega) = \frac{1}{T} \sum_{i=-\infty}^{\infty} R_X(i) \int_{-\infty}^{\infty} h^*(\alpha) \left[ \int_{-\infty}^{\infty} h(\alpha-iT+\tau) e^{-j\omega\tau} d\tau \right] d\alpha. \quad (3.152)$$

The expression in brackets is the Fourier transform of  $h(t)$  with a time shift of  $\alpha-iT$ ,

so it equals  $e^{j\omega(\alpha-iT)}H(j\omega)$ . Therefore,

$$S_Z(j\omega) = \frac{1}{T}H(j\omega) \sum_{i=-\infty}^{\infty} R_X(i) \left[ \int_{-\infty}^{\infty} h^*(\alpha) e^{j\omega(\alpha-iT)} d\alpha \right].$$

The expression in brackets is  $e^{-j\omega iT} H^*(j\omega)$ , getting

$$S_Z(j\omega) = \frac{1}{T}H(j\omega)H^*(j\omega) \sum_{i=-\infty}^{\infty} R_X(i)e^{-j\omega iT}. \quad (3.153)$$

The summation is simply the discrete-time Fourier transform  $S_X(e^{j\omega T})$  of the autocorrelation function. The final result is

$$S_Z(j\omega) = \frac{1}{T} |H(j\omega)|^2 S_X(e^{j\omega T}). \quad (3.154)$$

### APPENDIX 3-B POWER SPECTRUM OF A MARKOV CHAIN

In this appendix we solve the problem of finding the power spectrum of the random process (3.95). The power spectrum only exists if the random process is wide sense stationary. Strictly speaking, this requires that the Markov chain be running over all time, although we can interpret the results as indicative of the power spectrum for a chain that was initialized but has been running long enough to be in the steady-state. We approach this by assuming that the initial probability of each state is the same as its steady-state probability, so that the state probability is in fact constant with time (a *stationary* Markov chain).

We first determine the autocorrelation function of (3.95),

$$R_X(n) = E[f(\Psi_k)f(\Psi_{k+n})], \quad (3.155)$$

assuming  $f(\cdot)$  is a real-valued function. Assuming wide-sense stationarity, we can take  $k = 0$  and this can be written

$$R_X(n) = \sum_{i \in \Omega_\Psi} \sum_{j \in \Omega_\Psi} f(i)f(j)p_{0,n}(i,j). \quad (3.156)$$

where by Bayes' rule

$$p_{0,n}(i,j) = p_{n|0}(j|i)p_0(i) \quad (3.157)$$

is the joint probability of being in state  $i$  at time 0 and state  $j$  at time  $n$ . Assuming we have already calculated the steady-state state probabilities  $p(i)$  for the chain, by the stationarity assumption we can write

$$p(i) = p_0(i). \quad (3.158)$$

One way to think of this is as forcing the initial state probability to equal the steady-state probability, thus suppressing any transient solution. Finally, we must carefully

note the d.c. component of the random process, since it contributes a delta-function to the power spectrum that can easily be lost if we are not careful. Specifically, the d.c. component is

$$\mu_X = \sum_{i \in \Omega_\Psi} f(i)p(i). \quad (3.159)$$

The power spectrum is simply the Z transform of the autocorrelation function evaluated at  $z = e^{j\omega T}$  (see (3.58)). Rather than calculate the Z transform  $S_X(z)$  directly, let us first concentrate on the quantity

$$S_X^+(z) = \sum_{n=0}^{\infty} R_X(n)z^{-n} \quad (3.160)$$

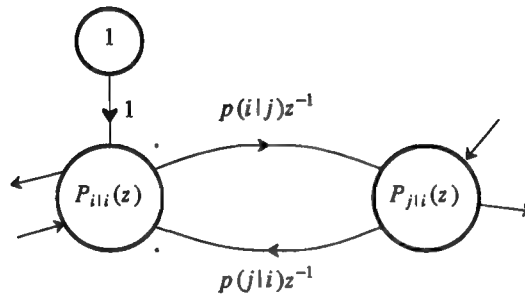
that includes only the positive index terms in the summation making up the Z transform. From (3.156), (3.157), and (3.158), this can be written as

$$S_X^+(z) = \sum_{i \in \Omega_\Psi} \sum_{j \in \Omega_\Psi} f(i)f(j)p(i) \cdot P_{j|i}(z) \quad (3.161)$$

where

$$P_{j|i}(z) = \sum_{n=0}^{\infty} p_{n|0}(j|i)z^{-n}. \quad (3.162)$$

This latter quantity can be interpreted as the Z-transform of  $p_{n|0}(j|i)$ , which is in turn the probability of being in state  $j$  at time  $n$  given that we started (with probability one) in state  $i$  at time 0. This quantity is easy to calculate using the techniques we have previously displayed, since it is simply the Z-transform of a transient solution starting with probability one in a particular state. The signal flow graph for this solution is shown in Figure 3-19, where only the states  $i$  and  $j$  are shown. This signal flow graph must be solved for  $P_{j|i}(z)$  for all  $(i,j)$  for which  $f(i)f(j)$  is non-zero in (3.161).



**Figure 3-19.** Signal flow graph representation of equations that must be solved to find  $P_{j|i}(z)$ .

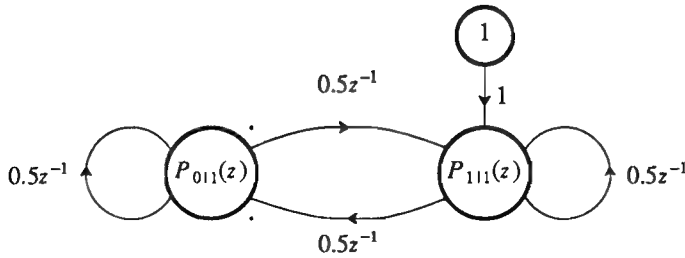


Figure 3-20. Signal flow graph for the parity check circuit.

#### Example 3-24.

Again returning to the parity check circuit of Example 3-14, let us compute  $S_X^+(z)$ . In this case  $f(i) = i$ , so that the random process  $X_k = f(\Psi_k) = \Psi_k$  assumes the values 0 and 1. For that case, we only need evaluate one term in (3.156), corresponding to  $i = j = 1$ , and all the others are zero. This term is shown by the signal flow graph in Example 3-24. Solving this flow graph, we get

$$P_{111}(z) = \frac{1 - 0.5z^{-1}}{1 - z^{-1}} \quad (3.163)$$

and

$$S_X^+(z) = 0.5 \frac{1 - 0.5z^{-1}}{1 - z^{-1}} \quad (3.164)$$

since there is only one term in the sum and  $p(0) = p(1) = 1/2$ . Inverting the Z transform, we find that

$$R_X(n) = \begin{cases} 1/2; & \text{for } n = 0 \\ 1/4; & \text{for } n > 0 \end{cases} \quad (3.165)$$

This result says that the power of the process is  $1/2$ , which is obvious, and that the process has a d.c. component of  $1/2$  since the autocorrelation function approaches  $1/4$  for large  $n$ , which is also obvious.  $\square$

We have determined the one-sided terms in the power spectrum, and we must generate the two-sided spectrum  $S_X(z)$ . However, before doing this, we must first remove any d.c. component, since that d.c. component can be represented by the one-sided transform but is problematic in the two-sided transform. This is simple, since we only need to replace  $S_X^+(z)$  by

$$S_X^+(z) - \frac{\mu_X^2}{1 - z^{-1}} \quad (3.166)$$

to remove this d.c. component. Alternatively we could have defined a new random process with the d.c. component removed, although that method is often harder.

**Example 3-25.**

For the parity check circuit of Example 3-14, the d.c. component is  $\mu_X = 1/2$ , and subtracting the appropriate term from (3.164),

$$S_X^+(z) - \frac{\mu_X^2}{1-z^{-1}} = 0.25. \quad (3.167)$$

Note that for this process this result would have been much more difficult to obtain if we had defined a d.c. free random process, since then we would have to evaluate all four terms in (3.161) rather than just one.  $\square$

We must now turn the one-sided version of the power spectrum into a two-sided version. The Z transform of the autocorrelation function can be written

$$S_X(z) = \sum_{m=-\infty}^{\infty} R_X(m)z^{-m} = \sum_{m=0}^{\infty} R_X(m)z^{-m} + \sum_{m=0}^{\infty} R_X(m)z^m - R_X(0), \quad (3.168)$$

where we have used the symmetry of the autocorrelation function. Noting that  $R_X(0) = S_X^+(\infty)$ , we get finally

$$S_X(z) = S_X^+(z) + S_X^+(z^{-1}) - S_X^+(\infty). \quad (3.169)$$

**Example 3-26.**

To finish with the parity check example of Example 3-14,

$$S_X(z) = 0.25 + 0.25 - 0.25 = 0.25 \quad (3.170)$$

and the process is white with power  $1/4$ . However, recall that this power spectrum does not include the d.c. term, so that in fact

$$S_X(e^{j\omega T}) = \frac{1}{4} + \frac{\pi}{2} \delta(\omega). \quad (3.171)$$

The area of the delta function has been chosen so that this area divided by  $2\pi$  is  $1/4$ , the power of the d.c. component.  $\square$

## APPENDIX 3-C DERIVATION OF POISSON PROCESS

In this appendix we show that the Poisson distribution for the accumulated number of arrivals as given by (3.134) is valid. To begin with, we need the solution to a first-order differential equation, which is given in the following exercise[13].

**Exercise 3-27.**

Consider the following first order differential equation,

$$\dot{x}(t) + a(t)x(t) = b(t). \quad (3.172)$$

- (a) Let  $\dot{A}(t) = a(t)$  and show that

$$\frac{d}{dt}(e^{\Lambda(t)} x(t)) = b(t) e^{\Lambda(t)}. \quad (3.173)$$

(b) Integrate both sides of (3.173) to obtain the solution for  $x(t)$

$$x(t) = x(t_0) e^{\Lambda(t)} + e^{-\Lambda(t)} \int_{t_0}^t b(u) e^{\Lambda(u)} du, \quad \Lambda(t) = \int_{t_0}^t a(v) dv. \quad (3.174)$$

□

Returning to the Poisson process, identify

$$a(t) = \lambda(t), \quad b(t) = \lambda(t) q_{j-1}(t). \quad (3.175)$$

Therefore, given the definition of (3.133) for  $\Lambda(t)$ ,

$$q_j(t) = q_j(t_0) e^{-\Lambda(t)} + e^{-\Lambda(t)} \int_{t_0}^t \lambda(u) q_{j-1}(u) e^{\Lambda(u)} du. \quad (3.176)$$

The solution follows immediately for  $j = 0$  using the initial condition of (3.132),

$$q_0(t) = e^{-\Lambda(t)} \quad (3.177)$$

and the rest is easy!

#### Exercise 3-28.

Verify the validity of (3.134) by induction on (3.176). □

## APPENDIX 3-D MOMENT GENERATING FUNCTION OF SHOT NOISE

In this appendix we derive the moment generating function of a shot noise process  $X(t)$  corresponding to impulse response  $h(t)$ . A sample function of such a process is given by (3.136).

To find the moment generating function, divide the time axis into small intervals of length  $\delta t$ , where the  $k$ -th interval is  $[(k - 1/2) \cdot \delta t, (k + 1/2) \cdot \delta t]$ . Group all the arrivals in the  $k$ -th interval together into a single impulse of height  $N_k$  located at time  $k \cdot \delta t$ , where  $N_k$  is the number of arrivals in the  $k$ -th interval. Thus, the shot noise of (3.136) becomes approximately

$$X(t) = \sum_{k=-\infty}^{\infty} N_k h(t - k \cdot \delta t) \quad (3.178)$$

where this equation becomes increasingly accurate as  $\delta t \rightarrow 0$ .

Since the intervals are non-overlapping, the  $N_k$  are independent Poisson random variables with parameter  $\lambda(k \cdot \delta t) \cdot \delta t$ , the average number of arrivals in the interval. The moment generating function of  $N_k$  is therefore

$$\log_e \Phi_{N_k}(s) = \lambda(k \cdot \delta t) \cdot \delta t (e^s - 1) \quad (3.179)$$

and the moment generating function of (3.178) is

$$\begin{aligned} \Phi_{X(t)}(s) &= E[\exp\{s \sum_{k=-\infty}^{\infty} N_k h(t - k \cdot \delta t)\}] = \prod_{k=-\infty}^{\infty} E[\exp\{s N_k h(t - k \cdot \delta t)\}] \\ &= \prod_{k=-\infty}^{\infty} \Phi_{N_k}(s h(t - k \cdot \delta t)) . \end{aligned} \quad (3.180)$$

Taking the logarithm of the moment generating function, and substituting from (3.179),

$$\log_e \Phi_{X(t)} = \sum_{k=-\infty}^{\infty} \lambda(k \cdot \delta t) (\exp\{s h(t - k \cdot \delta t)\} - 1) \cdot \delta t \quad (3.181)$$

and as  $\delta t \rightarrow 0$  this approaches the integral

$$\log_e \Phi_{X(t)} = \int_{-\infty}^{\infty} \lambda(\tau) (\exp\{s h(t - \tau)\} - 1) d\tau \quad (3.182)$$

which we recognize as the convolution of (3.137).

## PROBLEMS

- 3-1. Use the moment generating function of (3.41) to show that the mean of the Gaussian distribution is  $\mu$  and the variance  $\sigma^2$ .
- 3-2. Show that the marginal p.d.f.s of  $X$  and  $Y$  in (3.47) are those of a zero-mean Gaussian random variable with variance  $\sigma^2$ .
- 3-3. Show that for  $y > 0$

$$\frac{1}{y\sqrt{2\pi}} e^{-y^2/2} \left[ 1 - \frac{1}{y^2} \right] < Q(y) < \frac{1}{y\sqrt{2\pi}} e^{-y^2/2} . \quad (3.183)$$

These bounds are plotted in Figure 3-1. **Hint:** Write the definition of  $Q(\cdot)$  from (3.38) and integrate by parts.

- 3-4. Let  $X$  and  $Y$  be two complex-valued random variables.
  - (a) Form an estimate of  $X$  as  $\hat{X} = a \cdot Y$  for some complex number  $a$ . Find the  $a$  that minimizes the mean-square error  $E[|\hat{X} - X|^2]$ .
  - (b) Reformulate the problem of (a) in terms of linear space and inner products.
  - (c) Re-solve the problem of (a) using the projection theorem of Section 2.6.3.
- 3-5. In Figure 3-4a let  $E_k$  be a prediction error generated by filter  $E(z)$  such that

$$E[E_{k+m} X_k^*] = R_{EX}(m) = 0, \quad m > 0, \quad (3.184)$$

and let  $E_k'$  be the output generated by any other causal and monic filter.

- (a) Show that

$$E|E_k'|^2 = E|E_k' - E_k|^2 + E|E_k|^2, \quad (3.185)$$

thus establishing that the output MSE is minimized when  $E_k' = E_k$ .

- (b) Show that it follows from the orthogonality property of (3.184) that  $R_E(m) = 0$  for all  $m \neq 0$ , and hence the optimal prediction error must be white.

**3-6.**

- (a) Restate the results of Problem 3-5 in geometric terms, using the interpretation of Section 3.1.4.  
 (b) Re-derive the results of Problem 3-5 using the projection theorem of Section 2.6.3.

- 3-7.** Given a WSS random process  $X(t)$  with power  $R_X(0)$ , show that the sampled random process  $Y_k = X(kT)$  has the same power,

$$E[|Y_k|^2] = R_Y(0) = R_X(0). \quad (3.186)$$

- 3-8.** Given a sequence of i.i.d. random variables  $A_k$  which take on values  $\pm 1$  with equal probability, find an expression for  $E[A_p A_q A_r A_s]$ .

- 3-9.** Consider a random process  $X(t)$  filtered by an ideal bandpass filter with frequency response

$$H(j\omega) = \begin{cases} 1; & \omega_a < \omega < \omega_b \\ 0; & \text{otherwise} \end{cases}$$

Let  $Y(t)$  be the output of the filter. Show that

$$R_Y(0) = \frac{1}{2\pi} \int_{\omega_a}^{\omega_b} S_X(j\omega) d\omega.$$

Use this to show that  $S_X(j\omega) \geq 0$  for all  $\omega$ .

- 3-10.** Extending Exercise 2-6 to random signals, assume the input to the possibly complex-valued LTI system shown in Figure 2-3 is a wide sense stationary complex-valued discrete-time random process with power spectral density  $S_X(e^{j\omega T}) = N_0$ . Show that the autocorrelation of the output is

$$R_Y(k) = N_0 f(kT) * f^*(-kT) = N_0 \sum_m f(mT) f^*((m-k)T) \quad (3.187)$$

- 3-11.** Show that the cross-correlation function has symmetry

$$R_{XY}(\tau) = R_{YX}^*(-\tau). \quad (3.188)$$

Is the cross-spectral density of two random processes necessarily real-valued?

- 3-12.** Where a Markov chain has unique steady-state probabilities  $p_k(i) = p(i)$ , they can be found from the condition that the state probabilities will not change with one time increment. Assume  $\Omega_\psi = \{0, \dots, M\}$ , define the matrix of state transition probabilities  $\mathbf{P}$  to contain  $p(j|i)$  in its  $(i, j)^{th}$  entry, and define the vector  $\pi = [p(0), \dots, p(M)]$  to contain the steady-state probabilities, if they exist. Show that the steady-state probabilities can be obtained by solving the system of equations  $\pi = \pi\mathbf{P}$  with the constraint

$$\sum_{i=0}^M p(i) = 1. \quad (3.189)$$

- 3-13.** Assume you toss a coin that is not fair, where  $p$  is the probability of a tail and  $q = 1 - p$  is the probability of a head.

- (a) Draw a signal flow graph representation for a Markov chain representing the number of heads tossed in a row. Define  $N$  as an absorption state, since in part (c) we will be interested in the first passage time to state  $N$ .  
 (b) Show that

$$P_N(z) = \frac{q^N z(z - q)}{(z - 1)(z^{N+1} - z^N + pq^N)}. \quad (3.190)$$



- (c) Show that the first passage time to  $N$  heads in a row is

$$f_N = \frac{1 - q^N}{pq^N}. \quad (3.191)$$

- (d) Interpret this equation for  $p \approx 1$  and  $N$  large.

- 3-14. Show that for a Markov chain  $\Psi_k$ ,

$$p(\psi_0, \psi_1, \dots, \psi_n) = p(\psi_n | \psi_{n-1})p(\psi_{n-1} | \psi_{n-2}) \cdots p(\psi_1 | \psi_0)p(\psi_0).$$

In words, show that the joint probability of the states at times zero through  $n$  is the product of the initial state probability  $p(\psi_0)$  and the transition probabilities  $p(\psi_k | \psi_{k-1})$ .

- 3-15. Show that for a Markov chain  $\Psi_k$

$$p(\psi_n | \psi_{n+1}, \psi_{n+2}, \dots, \psi_{n+m}) = p(\psi_n | \psi_{n+1}). \quad (3.192)$$

In words, show that a Markov chain is also Markov when time is reversed.

- 3-16. Show that for the Markov chain  $\Psi_k$ , the future is independent of the past if the present is known. In other words, for any  $n > r > s$ ,

$$p(\psi_n, \psi_s | \psi_r) = p(\psi_n | \psi_r)p(\psi_s | \psi_r).$$

- 3-17. Consider the parity checker example in Figure 3-8. Suppose that the initial state is zero,  $p_0(0) = 1$ . Sketch the signal flow graph describing the state probabilities. Compute  $p_k(0)$  and  $p_k(1)$  as a function of  $k$ . Sketch these functions. Is the Markov chain stationary?

- 3-18. Consider tossing a fair coin. We are interested in the probability that at the  $k^{\text{th}}$  toss we have seen at least two heads in a row. Define the random process  $\Psi_k$  to have value two if there have been two heads in a row, to have value one if not and the last toss was heads, and to have value zero otherwise.

- Show that the random process  $\Psi_k$  is Markov and sketch the state diagram of the Markov chain.
- Sketch the signal flow graph describing the state probabilities. Assume that the coin is fair.
- Solve for the probability that at the  $k^{\text{th}}$  toss we have seen at least two heads in a row. You may leave the solution in the  $Z$  domain.

- 3-19. Using the results of Exercise 3-3, show that the Chernoff bounds on the distribution function for a Poisson random variable  $N$  with parameter  $a$  are

$$1 - F_N(n) \leq \left(\frac{a}{n}\right)^n e^{n-a}, \quad a < n, \quad F_N(n) \leq \left(\frac{a}{n}\right)^n e^{n-a}, \quad a > n. \quad (3.193)$$

- 3-20. Find the mean and variance at time  $t_1$  of a Poisson process  $N(t)$  with constant rate  $\lambda$ .

- 3-21. Show that if  $t_1 < t_2$  then

$$p_{N(t_1), N(t_2)}(k, k+n) = \frac{\lambda^{k+n} (t_2 - t_1)^n t_1^k}{n! k!} e^{-\lambda t_2}.$$

- 3-22. Consider a pure birth process in which the birth rate is proportional to the state ( $\lambda_j(t) = j\lambda$ ), as might model the growth of a biological population. Assume the initial condition is  $q_1(0) = 1$ , that is we start with a population of one. Find  $Q_j(s)$  for all  $j$ .

- 3-23. Shot noise can be generated from a Poisson process by linear filters as shown in Figure 3-21. Assume without further justification that expectation and differentiation can be interchanged; that is, the mean value of  $\frac{dN(t)}{dt}$  is  $\frac{d}{dt}E[N(t)]$ .

- For  $N(t)$  a Poisson process, show that the mean value of  $\frac{dN(t)}{dt}$  is  $\lambda(t)$ .

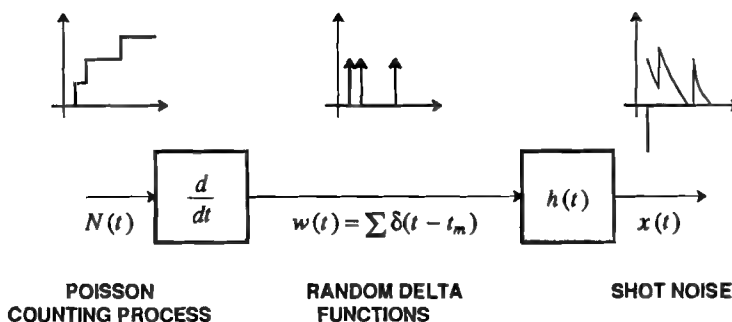


Figure 3-21. The generation of the shot noise from a Poisson counting process.

- (b) Similarly show that the mean value of  $X(t)$  is given by (3.138).  
 (c) For a random process  $N(t)$ , show that the derivative of this process  $\dot{N}(t)$  has autocorrelation

$$R_{\dot{N}\dot{N}}(t_1, t_2) = \frac{\partial^2 R_{NN}(t_1, t_2)}{\partial t_1 \partial t_2}. \quad (3.194)$$

- (d) Consider a linear time-invariant system with input  $W(t)$  and output  $X(t)$ , where  $W(t)$  has autocorrelation function  $R_{WW}(t_1, t_2)$ . Show that

$$R_{WX}(t_1, t_2) = R_{WW}(t_1, t_2) * h(t_2), \quad R_{XX}(t_1, t_2) = R_{WX}(t_1, t_2) * h(t_1). \quad (3.195)$$

- 3-24. For the Poisson process  $N(t)$  in Figure 3-21, consider two times  $0 < t_1 < t_2$ , and note the statistical independence of  $(N(t_1) - N(0))$  and  $(N(t_2) - N(t_1))$ . Using this fact, and assuming  $N(0) = 0$ , show that

$$R_{NN}(t_1, t_2) = \Lambda(t_1)[1 + \Lambda(t_2)], \quad t_1 \leq t_2 \quad (3.196)$$

where  $\Lambda(t)$  is defined in (3.133). Exchange the role of  $t_1$  and  $t_2$  to show that

$$R_{NN}(t_1, t_2) = \Lambda(t_2)[1 + \Lambda(t_1)], \quad t_1 \geq t_2. \quad (3.197)$$

- 3-25. Using the results of Problem 3-23 and Problem 3-24, show that the autocorrelation of shot noise is

$$R_X(t_1, t_2) = [\lambda(t_1) * h(t_1)][\lambda(t_2) * h(t_2)] + [\lambda(t_2)h(t_1 - t_2)] * h(t_2), \quad (3.198)$$

and evaluating at  $t_1 = t_2 = t$ ,

$$R_X(t, t) = [\lambda(t) * h(t)]^2 + \lambda(t) * h^2(t) \quad (3.199)$$

thereby establishing Campbell's theorem (3.139) by a different method.

- 3-26. For the constant rate case ( $\lambda(t) = \lambda$ ), the shot noise process is wide-sense stationary. Find the autocorrelation and power spectrum.

- 3-27. Let a Poisson process have rate

$$\lambda(t) = \begin{cases} 0, & t < 0 \\ \lambda_0, & t \geq 0 \end{cases}$$

Show that a shot noise with this rate has mean value proportional to the step function of the system.

- 3-28. Consider a shot noise with rate function

$$\lambda(t) = \lambda_0 + \lambda_1 \cos(\omega_1 t).$$

Find the mean value of this shot noise.

- 3-29. Show that the power spectrum of the output of the parity checker of Figure 3-8 when the input bits are not equally probable is

$$S_X(z) = \frac{p(1-p)}{(1 - (1-2p)z^{-1})(1 - (1-2p)z)} \quad (3.200)$$

where  $p$  is the probability of a one-bit.

## REFERENCES

1. R. E. Ziemer and W. H. Tranter, *Principles of Communications: Systems Modulation and Noise*, Houghton Mifflin Co., Boston (1985).
2. S. J. Mason, "Feedback Theory - Some Properties of Signal Flow Graphs," *Proc. IEEE* **41**(Sep. 1953).
3. S. J. Mason, "Feedback Theory — Further Properties of Signal Flow Graphs," *Proc. IRE* **44**(7) p. 920 (July 1956).
4. B. C. Kuo, *Automatic Control Systems*, Prentice-Hall, Englewood Cliffs, N.J. (1962).
5. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, Prentice-Hall, Englewood Cliffs, N.J. (1988).
6. R. B. Cooper, *Introduction to Queueing Theory*, MacMillan, New York (1972).
7. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York (1991).
8. H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*, Prentice-Hall, Englewood Cliffs, NJ (1986).
9. S. M. Ross, *Stochastic Processes*, John Wiley & Sons, New York (1983).
10. Sheldon M. Ross, *Introduction to Probability Models, 2nd Ed.*, Academic Press, New York (1980).
11. J. F. Hayes, *Modeling and Analysis of Computer Communication Networks*, Plenum Press, New York (1984).
12. L. Kleinrock, *Queueing Systems. Volume I: Theory*, John Wiley & Sons, New York (1975).
13. E. A. Coddington, *An Introduction to Ordinary Differential Equations*, Prentice Hall, Englewood Cliffs, N.J. (1961).

# 4

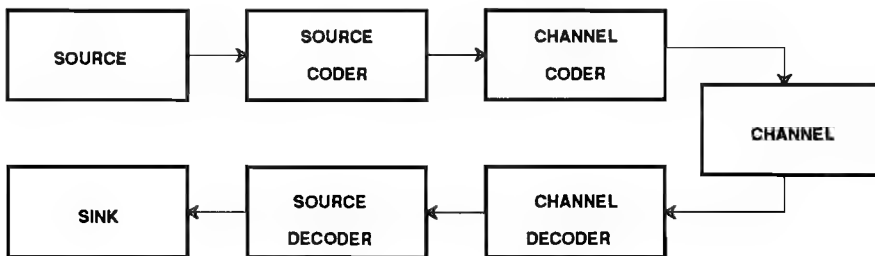
---

## LIMITS OF COMMUNICATION

---

In the late 1940's, Claude Shannon of Bell Laboratories developed a mathematical theory of information that profoundly altered our basic thinking about communication, and stimulated considerable intellectual activity, both practical and theoretical. This theory, among other things, gives us some fundamental boundaries within which communication can take place. Often we can gain considerable insight by comparing the performance of a digital communication system design with these limits.

Information theory provides profound insights into the situation pictured in



**Figure 4-1.** A general picture of a source communicating over a channel using source and channel coding.

Figure 4-1, in which a *source* is communicating over a *channel* to a *sink*. The source and channel are both modeled statistically. The objective is to provide the source information to the sink with the greatest fidelity. To that end, Shannon introduced the general idea of *coding*. The objective of *source coding* is to minimize the bit rate required for representation of the source at the output of a source coder, subject to a constraint on fidelity. Shannon showed that the interface between the source coder and channel coder can be, without loss of generality, a bit stream, regardless of the nature of the source and channel. The objective of *channel coding* is to maximize the information rate that the channel can convey sufficiently reliably (where reliability is normally measured as a bit error probability). Our primary focus in this book will be on the channel and the associated channel coder, although understanding source coding will also be helpful.

Given the statistics of a source, modeled as a discrete-time random process, the minimum number of bits per unit time required to represent it at the output of the source coder with some specified distortion can be determined. The *source coding theorem* is the key result of this *rate distortion theory* (see for example [1]). This theory offers considerable insight into the bit rates required for digital communication of an analog signal via PCM (Chapter 1).

#### Example 4-1.

We limit our attention here to the simple special case of a *discrete-time discrete-valued* random process  $\{X_k\}$  with independent and identically distributed (i.i.d.) samples. Because the process is discrete-valued, it is possible to encode the signal as a bit stream with *perfect fidelity*. In fact, the minimum average number of bits required to represent each sample without distortion is equal to the *entropy* of  $X$ , defined to be

$$H(X) = E[-\log_2 p_X(X)] = - \sum_{x \in \Omega_X} p_X(x) \log_2 p_X(x), \quad (4.1)$$

where  $\Omega_X$  is the alphabet (sample space) of  $X$ . This result is developed in Section 4.1.  $\square$

Since the entropy determines the number of bits required to represent a sample at the output of the source coder, it is said to determine the amount of *information* in the sample, measured in bits. This concept is explained in Section 4.1.

A second concept due to Shannon is the *capacity* of a noisy communication channel, defined as the maximum bit rate that can be transmitted over that channel with a vanishingly small error rate. The various forms of the *channel coding theorem* specify the capacity. The fact that an error rate approaching zero can be achieved was very surprising at the time, and it motivated the practical forms of channel coding to be discussed in Chapters 13 and 14.

#### Example 4-2.

Consider transmitting a random process  $\{X_k\}$ , with similar characteristics to Example 4-1, over a noisy discrete-time memoryless channel, defined as one for which the current output  $Y_k$  is dependent on only the current input  $X_k$ . Because the channel is memoryless, the samples  $Y_k$  are also independent and identically distributed. The capacity of this channel can be obtained from the *mutual information* between the input random variable  $X$  and the output random variable  $Y$ ,

$$I(X,Y) = H(X) - H(X|Y), \quad (4.2)$$

where  $H(X|Y)$  is the *conditional entropy*. The channel capacity equals the mutual information maximized over all possible probability distributions for the input  $X$ . This result is developed in Section 4.2.  $\square$

The result of Example 4-2 can also be used to determine the channel capacity of a bandlimited continuous-time channel using the Nyquist sampling theorem, as will be discussed in Section 4.3.

## 4.1. JUST ENOUGH INFORMATION ABOUT ENTROPY

Intuitively, *observing the outcome* of a random variable gives us *information*. Rare events carry more information than common events.

### Example 4-3.

You learn very little if I tell you that the sun rose this morning, but you learn considerably more if I tell you that San Francisco was destroyed by an earthquake this morning. The reason the latter observation carries more information is that it has a lower prior probability.  $\square$

In 1928 Hartley proposed a logarithmic measure of information that reflects this intuition. Consider a random variable  $X$  with sample space  $\Omega_X = \{a_1, a_2, \dots, a_K\}$ . The *self-information* in an outcome  $a_m$  is defined to be

$$h(a_m) = -\log_2 p_X(a_m). \quad (4.3)$$

The self-information of a rare event is greater than the self-information of a common event, conforming with intuition. Furthermore, the self-information is non-negative. But why the logarithm? One intuitive justification arises from considering two independent random variables  $X$  and  $Y$ , where  $\Omega_Y = \{b_1, b_2, \dots, b_N\}$ . The information in the joint events  $a_m$  and  $b_n$  intuitively should be the sum of the information in each. The self information defined in (4.3) has this property,

$$\begin{aligned} h(a_m, b_n) &= -\log_2 p_{X,Y}(a_m, b_n) = -\log_2 p_X(a_m) - \log_2 p_Y(b_n) \\ &= h(a_m) + h(b_n). \end{aligned} \quad (4.4)$$

The *average information*  $H(X)$  in  $X$ , defined in (4.1), is also called the *entropy* of  $X$  because of its formal similarity to thermodynamic entropy. Equivalent interpretations of  $H(X)$  are

- the average information obtained by observing an outcome,
- the average uncertainty about  $X$  before it is observed, and
- the average uncertainty removed by observing  $X$ .

Because of the base-two logarithm in (4.1), information is measured in *bits*.

**Example 4-4.**

Consider a binary random variable  $X$  with alphabet  $\Omega_X = \{0,1\}$ . Suppose that  $q = p_X(1)$ , so

$$H(X) = -q \log_2 q - (1-q) \log_2 (1-q). \quad (4.5)$$

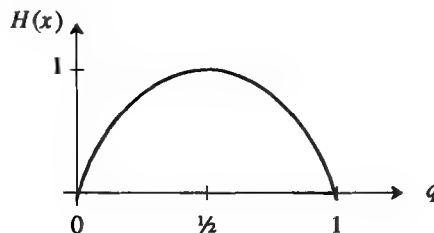
This is plotted as a function of  $q$  in Figure 4-2. Notice that the entropy peaks at 1 bit when  $q = 1/2$  and goes to zero when  $q = 0$  or  $q = 1$ . This agrees with our intuition that there is no information in certain events.  $\square$

Although the intuitive justification given so far may seem adequate, the key to the interpretation of entropy as an information measure lies in the *asymptotic equipartition theorem*, which is further justified in Appendix 4-A. Define the random vector  $\mathbf{X} = (X_1, \dots, X_n)$  where  $X_i$  are independent trials of a discrete random variable  $X$  with entropy  $H(X)$ . Define the vector  $\mathbf{x}$  to be an outcome of the random vector  $\mathbf{X}$ . The theorem says that asymptotically as  $n \rightarrow \infty$ , there is a set of "typical" outcomes  $S$  for which

$$p_X(\mathbf{x}) \approx 2^{-nH(X)}, \quad \mathbf{x} \in S, \quad (4.6)$$

and the total probability that the outcome is in  $S$  is very close to unity. Since the "typical" outcomes all have approximately the same probability, there must be approximately  $2^{nH(X)}$  outcomes in  $S$ . This approximation becomes more accurate as  $n$  gets large.

We can now conceptually design a source coder as follows. This source coder will assign to each outcome  $\mathbf{x}$  a binary word, called the *code*. If  $n$  is large, we can assign binary words only to the "typical" outcomes, and ignore the "nontypical" ones. If we use  $nH(X)$ -bit code words, we can encode each of the  $2^{nH(X)}$  typical outcomes with a unique binary word, for an average of  $H(X)$  bits per component of the vector  $\mathbf{x}$ . Since each outcome of the component random variable  $X$  requires on average  $H(X)$  bits,  $H(X)$  is the average information obtained from the observation. It is important to note, however, that this argument applies only if we encode a large number of components collectively, and not each component separately. The statement that  $H(X)$  is the average number of bits required to encode a component  $X$  applies only to an average of  $n$  components, not to an individual component.



**Figure 4-2.** The entropy of a binary random variable as a function of the probability  $q = p_X(1)$ .

We will now state (but not prove) the *source coding theorem* for discrete-amplitude discrete-time sources. If a source can be modeled as repeated independent trials of a random variable  $X$  at  $r$  trials per second, we define the *rate* of the source to be  $R = rH(X)$ . The source can be encoded by a source coder into a bit stream with bit rate less than  $R + \epsilon$  for any  $\epsilon > 0$ .

Constructing practical codes that come close to  $R$  is difficult, but constructing good sub-optimal codes is often easy.

#### Example 4-5.

For the source of Example 4-4, if  $q = 1/2$  then  $H(X) = 1$ . This implies that to encode repeated outcomes of  $X$  we need one bit per outcome, on average. In this case, this is also adequate for each sample, not just on average, since the source is binary. A source coder that achieves rate  $R$  just transmits outcomes of  $X$  unaltered.  $\square$

#### Example 4-6.

When  $q = 0.1$  in Example 4-4,

$$H(X) = -0.1 \log_2(0.1) - 0.9 \log_2(0.9) \approx 0.47, \quad (4.7)$$

implying that less than half a bit per outcome is required, on average. This is not so intuitive; however, there are coding schemes in which the average number of bits per outcome will be lower than unity but greater than 0.47. One simple coding scheme takes a pair of outcomes and assigns them bits according to the following table.

outcomes	bits
0,0	0
0,1	10
1,0	110
1,1	111

A bit stream formed by repeated trials can be easily decoded. The average number of bits produced by this coder is 0.645 bits per trial. But note that the pair of trials 1,1 requires three bits, or 1.5 bits per trial. This emphasizes that the entropy is an average quantity.  $\square$

#### Example 4-7.

Consider a particularly unfair coin that *always* comes up heads. Then

$$H(X) = 0, \quad (4.8)$$

using the identity  $0 \log_2 0 = 0$ . This says that no bits are required to specify the outcome, which is valid.  $\square$

#### Exercise 4-1.

It is clear from the definition of entropy that  $H(X) \geq 0$ . Use the inequality  $\log x \leq x - 1$  to show that

$$H(X) \leq \log_2 K, \quad (4.9)$$

where  $K$  is the size of the alphabet of  $X$ , with equality if and only if the outcomes of  $X$  are equally likely.  $\square$



The conclusion of Exercise 4-1 is that  $\log_2 K$  bits *always* suffices to specify the outcomes, as is obvious since  $2^{\log_2 K} = K$  possible outcomes can be encoded by a straightforward assignment, at least when  $K$  is a power of two. The less obvious conclusion is that the maximum number of bits,  $\log_2 K$ , is *required* only when the outcomes are equally likely.

## 4.2. CAPACITY OF DISCRETE-TIME CHANNELS

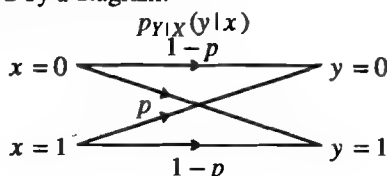
The concept of entropy and information can be extended to channels, yielding considerable information about their fundamental limits. This section considers discrete-time channels, deferring continuous-time channels to Section 4.3. We consider three different types of discrete-time channels: discrete-valued inputs and outputs, discrete-valued inputs and continuous-valued outputs, and continuous-valued inputs and outputs.

### 4.2.1. Discrete-Valued Inputs and Outputs

Consider a *discrete-time* channel with input random process  $\{X_k\}$  and output  $\{Y_k\}$ . We consider here only *memoryless channels* for which the current output  $Y_k$  is independent of all inputs except  $X_k$ . Such a channel is fully characterized by the conditional probabilities  $p_{Y|X}(y|x)$  for all  $x \in \Omega_X$  and  $y \in \Omega_Y$ .

#### Example 4-8.

Consider a channel with input and output alphabet  $\Omega_X = \Omega_Y = \{0,1\}$  such that  $p_{Y|X}(0|1) = p_{Y|X}(1|0) = p$ . This *binary symmetric channel (BSC)* offers a useful model of a channel that introduces independent random errors with probability  $p$ . The transition probabilities may be illustrated by a diagram:



□

If the input samples are independent, the information per sample at the input is  $H(X)$  and the information per  $n$  samples is  $nH(X)$ . The question is how much of this information gets through the channel. We can answer this question by finding the uncertainty in  $X$  after observing the output of the channel  $Y$ . Suppose that  $y$  is an outcome of  $Y$ . Then the uncertainty in  $X$  given the event  $Y = y$  is

$$H(X|y) = E \left[ -\log_2 p_{X|Y}(X|y) \right] = - \sum_{x \in \Omega_X} p_{X|Y}(x|y) \log_2 p_{X|Y}(x|y). \quad (4.10)$$

To find the average uncertainty in  $X$  after observing  $Y$ , we must average this over the distribution of  $Y$ , yielding a quantity called the *conditional entropy*,

$$H(X|Y) = \sum_{y \in \Omega_Y} H(X|y) p_Y(y) = - \sum_{y \in \Omega_Y} \sum_{x \in \Omega_X} p_{X,Y}(x,y) \log_2 p_{X|Y}(x|y). \quad (4.11)$$

This conditional entropy, on a channel such as the BSC, is a measure of the average uncertainty about the input of the channel after observing the output.

The uncertainty about  $X$  must be larger before observing  $Y$  than after; the difference is a measure of the information passed through the channel on average. Thus we define

$$I(X,Y) = H(X) - H(X|Y) \quad (4.12)$$

as the *average mutual information* (as in (4.2)). In other words,  $I(X,Y)$  is interpreted as the uncertainty about  $X$  that is removed by observing  $Y$ , or the information about  $X$  in  $Y$ .

#### Exercise 4-2.

- (a) Show that  $I(X,Y)$  can be written directly in terms of the transition probabilities (channel) and the input distribution (input) as

$$I(X,Y) = \sum_{x \in \Omega_X} p_X(x) \sum_{y \in \Omega_Y} p_{Y|X}(y|x) \log_2 \left[ \frac{p_{Y|X}(y|x)}{\sum_{x \in \Omega_X} p_X(x) p_{Y|X}(y|x)} \right]. \quad (4.13)$$

- (b) Show that (4.12) can be written alternatively as

$$I(X,Y) = H(Y) - H(Y|X) = I(Y,X). \quad (4.14)$$

Thus, the information about  $X$  in  $Y$  is the same as the information about  $Y$  in  $X$ .  $\square$

The transition probabilities are fixed by the channel. The input probabilities are under our control through the design of the channel coder. The mutual information (information conveyed through the channel) is a function of both transition and input probabilities. It makes intuitive sense that we would want to choose the input probabilities so as to maximize this mutual information. The *channel capacity per symbol* is defined as the maximum information conveyed over all possible input probability distributions,

$$C_s = \max_{p_X(x)} I(X,Y). \quad (4.15)$$

This capacity is in bits/symbol, where a symbol is one sample of  $X$ . If the channel is used  $s$  times per second, then the channel capacity in bits per second is

$$C = sC_s. \quad (4.16)$$

#### Exercise 4-3.

For the BSC of Example 4-8, let the probability of the two inputs be  $q$  and  $1 - q$ .

- (a) Show that the mutual information is

$$I(X,Y) = H(Y) + p \log_2 p + (1 - p) \log_2 (1 - p). \quad (4.17)$$

(b) By maximizing over  $q$ , show that the channel capacity per symbol is

$$C_s = 1 + p \log_2 p + (1 - p) \log_2 (1 - p). \quad (4.18)$$

The capacity is zero if  $p = 1/2$ , since then the channel inputs and outputs are independent, and is unity when  $p = 0$  or  $p = 1$ , since then the channel is binary and noiseless.  $\square$

Using the channel capacity theorem and the source coding theorem, we will now state (but not prove) a general *channel capacity theorem*. Given a source with rate  $R = rH(X)$  bits/second, and a channel with capacity  $C = sC_s$  bits/sec, then if  $R < C$  there exists a combination of source and channel coders such that the source can be communicated over the channel with fidelity arbitrarily close to perfect. If the source is a bit stream, the channel coder can achieve *arbitrarily low probability of error* if the bit rate is below the channel capacity. In practice, achieving vanishingly small error probability requires arbitrarily large computational complexity and processing delay. Nevertheless, the channel capacity result is very useful as an ideal against which to compare practical modulation and coding systems.

## 4.2.2. Discrete Inputs and Continuous Outputs

Another useful channel model is a discrete-time channel with a discrete-valued input and a continuous-valued output.

### Example 4-9.

In an *additive noise channel*, the output is

$$Y = X + N \quad (4.19)$$

where  $X$  is a discrete random input to the channel and  $N$  is a continuous noise variable. This model arises often in this book in the situation where a discrete *data symbol* taking on a finite number of possible values is transmitted over a channel with additive Gaussian noise (i.e.  $N$  is Gaussian).  $\square$

This model is useful because most communications media (Chapter 5) have continuous-valued outputs, due to thermal noise, whereas digital signals are discrete-valued.

The previous definitions of entropy carry over to continuous-valued random variables, if we are careful about replacing summations with integrals. For example, the entropy of a continuous-valued random variable  $Y$  is defined as

$$H(Y) = E[-\log_2 f_Y(y)] = - \int_{\Omega_Y} f_Y(y) \log_2 f_Y(y) dy. \quad (4.20)$$

Just as with discrete-valued random variables, it is possible to bound the entropy of a continuous-valued random variable.

### Exercise 4-4.

Show that if  $Y$  has zero mean and variance  $\sigma^2$ , then

$$0 \leq H(Y) \leq \log_2(\sigma\sqrt{2\pi e}) \quad (4.21)$$

with equality if and only if  $Y$  is Gaussian. **Hint:** Show that

$$H(Y) \leq - \int_{-\infty}^{\infty} f_Y(y) \log_2 g(y) dy \quad (4.22)$$

for any probability density function  $g(y)$ , using the inequality  $\log x \leq x - 1$ . Then substitute a Gaussian p.d.f. for  $g(y)$ .  $\square$

It is important to note that we have constrained the variance of the random variable in this exercise. A different constraint would lead to a different bound; or, no constraint could lead to *unbounded* entropy.

The conditional entropy is a little trickier because it involves both discrete and continuous-valued random variables. Following the second expression in (4.11), we can define

$$H(Y|X) = \sum_{x \in \Omega_X} p_X(x) \int_{\Omega_Y} f_{Y|X}(y|x) \log_2 f_{Y|X}(y|x) dy. \quad (4.23)$$

#### Exercise 4-5.

Consider the additive Gaussian noise of Example 4-9. Show that  $H(Y|X) = H(N)$ . This result is intuitive, since after observing the outcome of  $X$ , the uncertainty in  $Y$  is precisely the entropy of the noise.  $\square$

The mutual information and capacity are defined as before, in (4.12) and (4.15).

#### Exercise 4-6.

Following (4.13), the mutual information can be written in terms of the channel transition probability  $f_{Y|X}(y|x)$  and the probability distribution of the input  $p_X(x)$ ,

$$I(X,Y) = \sum_{x \in \Omega_X} p_X(x) \int_{\Omega_Y} f_{Y|X}(y|x) \log_2 \frac{f_{Y|X}(y|x)}{\sum_{x \in \Omega_X} p_X(x) f_{Y|X}(y|x)} dy. \quad (4.24)$$

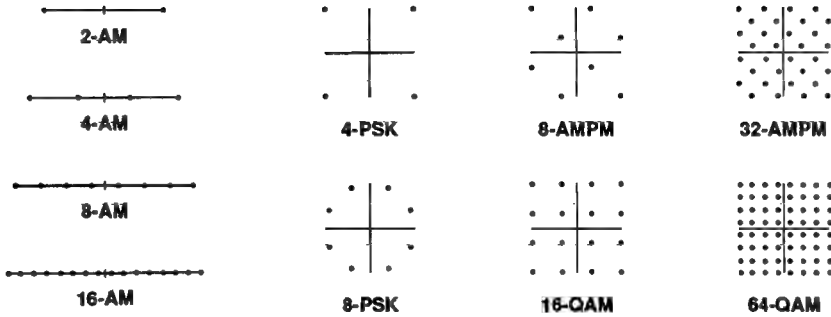
Derive this from (4.20) and (4.11).  $\square$

The channel capacity for the continuous-output channel depends on the values in the discrete input  $\Omega_X$ . For example, on an additive noise channel, we would expect the capacity of a channel with inputs  $\pm 100$  to be larger than the capacity with inputs  $\pm 1$  when the noise is the same. The set  $\Omega_X$  of channel inputs is called the input *alphabet*.

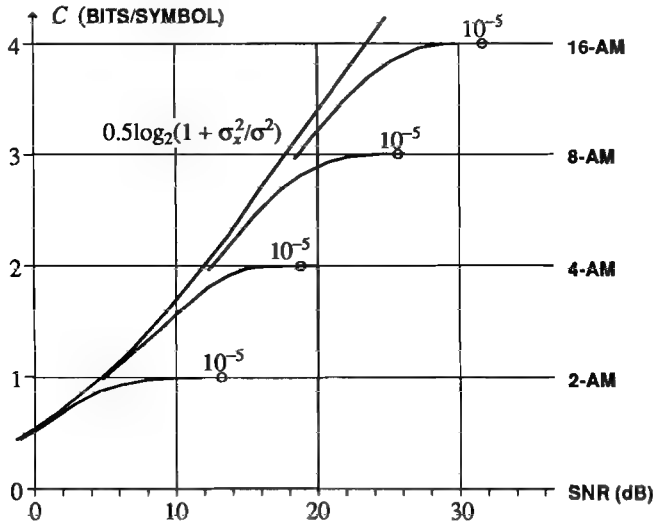
#### Example 4-10.

Some common channel alphabets that we will encounter in Chapter 6 are shown in Figure 4-3. The  $K$ -AM alphabets are real-valued, containing  $K$  equally spaced points centered at the origin. The remaining alphabets are complex-valued, as appropriate for complex-valued discrete-time channels. The noise in this case is assumed to be complex white Gaussian noise, where the real and imaginary parts have the same power but are independent of one another and of the channel input.  $\square$

One approach to calculating channel capacity would be not to constrain the alphabet at all; this is done in Section 4.2.3. Another approach is to choose an input alphabet,



**Figure 4-3.** Some real-valued and complex-valued channel alphabets for a discrete-valued channel input. The acronyms refer to signaling methods that will be discussed in Chapter 6.



**Figure 4-4.** Bounds on the information conveyed by a real-valued discrete-time channel with additive white Gaussian noise as a function of SNR for four input alphabets defined in Figure 4-3. It is assumed that the symbols in the alphabet are equally likely. Also shown is the channel capacity for continuous-valued input signals, derived in Section 4.2.3. The points labeled  $10^{-5}$  indicate the SNR at which a probability of error of  $10^{-5}$  is achieved with direct techniques (no coding). The significance of these points will be discussed further in Chapter 14. The variance of the transmitted symbols is  $\sigma_x^2$ , so the SNR is defined as  $\sigma_x^2/\sigma^2$ , and is expressed in dB. (After Ungerboeck [2].)

getting the discrete-input channel model of this subsection, and then determine the capacity by maximizing the mutual information over the probabilities of the inputs using (4.24). Going one step further, we can assume a particular distribution for the input alphabet, and then find the information  $I(X,Y)$  conveyed by the channel. In a classic paper that is credited with establishing the practical importance of *trellis*

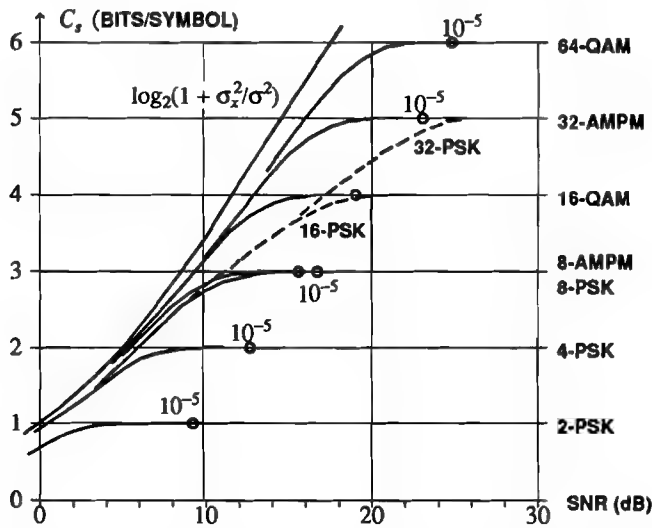
*coding* (Chapter 14), Ungerboeck makes this calculation assuming that the input symbols in the alphabet are equally likely and that the channel adds independent Gaussian noise [2]. He computes the information conveyed by the channel as a function of the *signal-to-noise ratio* (SNR) for the input alphabets in Figure 4-3. The results are shown in Figure 4-4 (real alphabets) and Figure 4-5 (complex alphabets).

**Example 4-11.**

Consider the curve corresponding to 4-AM. As the signal to noise ratio increases, the information conveyed approaches two bits per symbol. This is intuitive because if the noise is small, nearly two bits per symbol can be sent with an alphabet of four symbols with low probability of error. For each input alphabet  $\Omega_X$  with size  $|\Omega_X|$ , the information conveyed asymptotically approaches  $\log_2 |\Omega_X|$  as the signal to noise ratio increases. While a capacity of two bits per symbol is not achievable with 4-AM, it is achievable with 8-AM for an SNR as low as 13 dB. Furthermore, using 16-AM to transmit two bits per symbol does not gain much noise immunity. This suggests that there is very little lost if we use 8-AM to transmit two bits per symbol. This observation is exploited in Chapter 14, where we discuss trellis coding. □

**4.2.3. Continuous-Valued Inputs and Outputs**

The question arises as to what is lost by choosing a specific discrete alphabet at the channel input. We can answer this question by determining the capacity with a continuous-valued input, which is an infinite alphabet. For the additive Gaussian channel considered in Example 4-9, for any given SNR, we lose very little in capacity



**Figure 4-5.** An analog to Figure 4-4 for a discrete-time complex-valued alphabet (defined in Figure 4-3) and channel. (After Ungerboeck [ 2].)

by choosing a discrete input alphabet, as long as the alphabet is sufficiently large (the higher the SNR, the larger the required alphabet). This result is important in that it justifies many of the digital communication techniques used in practice (Chapter 6).

Let  $X$  be a continuous-valued random variable. The entropy of  $Y$  is still given by (4.20), but the summation over  $x$  in the conditional entropy (4.23) must be replaced by an integral,

$$H(Y|X) = \int_{\Omega_X} f_X(x) \int_{\Omega_Y} f_{Y|X}(y|x) \log_2 f_{Y|X}(y|x) dy. \quad (4.25)$$

We obtain the channel capacity by maximizing  $I(X, Y)$  over  $f_X(x)$ .

### Scalar Additive Gaussian Noise Channel

Assume an additive Gaussian noise channel,  $Y = X + N$  where  $N$  is an independent zero-mean Gaussian random variable with variance  $\sigma^2$ . What is the capacity under the constraint that the variance of  $X$  is  $\sigma_x^2$ ? The result of Exercise 4-5 is trivially extended to get  $H(Y|X) = H(N)$ , which is not a function of the input distribution, so the channel capacity is obtained by maximizing  $H(Y)$ . The variance of  $Y$  is constrained to be  $\sigma_x^2 + \sigma^2$ , so from (4.21),

$$H(Y) \leq \frac{1}{2} \log_2 [2\pi e (\sigma_x^2 + \sigma^2)], \quad (4.26)$$

with equality if and only if  $Y$  is Gaussian. Fortunately,  $Y$  is Gaussian if  $X$  is Gaussian, so the bound can in fact be achieved. Therefore channel capacity is achieved with a Gaussian input, and from (4.14),

$$C_s = \frac{1}{2} \log_2 [2\pi e (\sigma_x^2 + \sigma^2)] - \frac{1}{2} \log_2 (2\pi e \sigma^2) = \frac{1}{2} \log_2 \left(1 + \frac{\sigma_x^2}{\sigma^2}\right) \quad (4.27)$$

in bits per symbol. This channel capacity is plotted in both Figure 4-4 and Figure 4-5, where the SNR is  $\sigma_x^2/\sigma^2$ . Note that this capacity is very similar to the capacity for any particular discrete alphabet at low SNR, and diverges significantly at large SNR. The capacity in Figure 4-5 is twice that of Figure 4-4 because each of the real and imaginary parts has the capacity given by (4.27).

The conclusion is that for the Gaussian channel and any particular SNR, there is a sufficiently large discrete input alphabet that has a capacity very close to the continuous-input capacity. This result gives a solid theoretical underpinning to the practical use of discrete input alphabets, which are also very convenient for implementation (Chapter 6).

### Capacity of Vector Additive Gaussian Noise Channel

These results for the additive Gaussian channel are easily extended to a vector channel model. This extension will prove to be critically important in Chapters 8 and 10, where we consider continuous-time bandlimited Gaussian channels. We will show there that, for a given finite time interval, such a channel can be reduced to a vector Gaussian channel. Consider a channel modeled by

$$\mathbf{Y} = \mathbf{X} + \mathbf{N} \quad (4.28)$$

where  $\mathbf{X}$ ,  $\mathbf{Y}$ , and  $\mathbf{N}$  are  $N$ -dimensional vectors,  $\mathbf{X}$  and  $\mathbf{N}$  are independent, and the components of  $\mathbf{N}$  are independent Gaussian random variables each with variance  $\sigma^2$ . It is easily shown, as a generalization of Exercise 4-5, that

$$I(\mathbf{X}, \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) = H(\mathbf{Y}) - H(\mathbf{N}) \quad (4.29)$$

and that

$$H(\mathbf{N}) = \frac{N}{2} \log_2(2\pi e \sigma^2). \quad (4.30)$$

The entropy of a random vector is the same as that of a scalar random variable, (4.1) or (4.20), except that the sample space has vector-valued members. The noise entropy is proportional to the dimension  $N$  because each component of the noise contributes the same entropy as in the scalar case. All that remains, then, is to find the maximum of  $H(\mathbf{Y})$  over all input distributions  $f_{\mathbf{X}}(\mathbf{x})$ .

#### Exercise 4-7.

- (a) Generalize (4.21) to show that

$$H(\mathbf{Y}) \leq - \int_{\Omega_{\mathbf{Y}}} f_{\mathbf{Y}}(\mathbf{y}) \log_2 g(\mathbf{y}) d\mathbf{y} \quad (4.31)$$

for any probability density function  $g(\mathbf{y})$ .

- (b) Substitute a vector Gaussian density with independent components with mean zero and variance  $(\sigma^2 + \sigma_{x,n}^2)$  for the  $n$ -th component to obtain

$$H(\mathbf{Y}) \leq \frac{1}{2} \sum_{n=1}^N \log_2 [2\pi e (\sigma^2 + \sigma_{x,n}^2)], \quad (4.32)$$

and thus show that

$$I(\mathbf{X}, \mathbf{Y}) \leq \frac{1}{2} \sum_{n=1}^N \log_2 \left( 1 + \frac{\sigma_{x,n}^2}{\sigma^2} \right), \quad (4.33)$$

with equality if  $\mathbf{Y}$  is Gaussian with independent zero-mean components. Fortunately, this upper bound can be achieved if the input vector  $\mathbf{X}$  is chosen to have independent Gaussian components, each with mean zero and with variance  $\sigma_{x,n}^2$  for the  $n$ th component.

- (c) Using the inequality  $\log x \leq (x-1)$ , show that if the variance of  $\mathbf{X}$  is constrained to some  $\sigma_x^2$ ,

$$\sigma_x^2 = \sum_{n=1}^N \sigma_{x,n}^2, \quad (4.34)$$

then

$$I(\mathbf{X}, \mathbf{Y}) \leq \frac{N}{2} \log_2 \left( 1 + \frac{\sigma_x^2}{N\sigma^2} \right), \quad (4.35)$$

with equality if and only if all the components of  $\mathbf{X}$  have equal variance.  $\square$



The conclusion is that the capacity of the vector Gaussian channel with input variance constrained to  $E[|\mathbf{X}|^2] = \sigma_x^2$  is given by

$$C = \frac{N}{2} \log_2 \left( 1 + \frac{\sigma_x^2}{N\sigma^2} \right), \quad (4.36)$$

and the input distribution that achieves capacity is a zero-mean Gaussian vector with independent components, each with variance  $\sigma_x^2/N$ . The interpretation of this result is that the capacity is  $N$ , the number of degrees of freedom, times  $0.5 \cdot \log_2(1 + \text{SNR})$ , where the signal to noise ratio  $\text{SNR} = \sigma_x^2/N\sigma^2$  is the total input signal power divided by the total noise power.

### 4.3. FURTHER READING

Abramson [3] gives a short elementary introduction to information theory, particularly the channel coding theorem. Gallager [4] has long been a standard advanced text and includes an extensive discussion of continuous-time channels. McEliece [5] provides a readable introduction with qualitative sections devoted to describing the more advanced work in the field. An excellent recent text is by Cover and Thomas [6]. Also recommended is the text by Blahut [7]. A collection of key historical papers, edited by Slepian [8] provides an easy way to access the most important historical papers, including twelve by Shannon. "A Mathematical Theory of Communication" and "Communication in the Presence of Noise", two of Shannon's best known papers, are highly recommended reading, for their lucidity, relevance, and historical value. Especially interesting, and mandatory reading for anyone with an interest in the subject, Shannon gives an axiomatic justification of entropy as a measure of information. He simply assumes three properties that a reasonable measure of information should have, and derives entropy as the only measure that has these properties [9,10]. Viterbi and Omura [11] provide an encyclopedic coverage of information theory, with an emphasis throughout on convolutional codes. Finally, Wolfowitz [12] gives a variety of generalizations of the channel coding theorem.

## APPENDIX 4-A ASYMPTOTIC EQUIPARTITION THEOREM

In this appendix we give a non-rigorous derivation of the asymptotic equipartition theorem that gives a great deal of insight. Define a random process  $Y_k$  in which each sample is an independent trial of the random variable  $Y$  with alphabet  $\Omega_Y = \{b_1, \dots, b_K\}$ . Let there be  $n$  trials, and define  $n_i$  to be the number of outcomes equal to  $b_i$ . The relative-frequency interpretation of probabilities tells us that if  $n$  is large, then with high probability,

$$\frac{n_i}{n} \approx p_Y(b_i). \quad (4.37)$$

(A rigorous development depends mainly on defining precisely what we mean by "high probability". One approach is to show that given any  $\epsilon > 0$ , the probability that  $[p_Y(b_i) - \epsilon] < n_i/n < [p_Y(b_i) + \epsilon]$  approaches unity as  $n$  gets large.) Suppose that we are interested in the product of the  $n$  observations. We can write the product as

$$\begin{aligned} y_1 \cdots y_n &= (b_1)^{n_1} \cdots (b_K)^{n_K} = \left[ (b_1)^{n_1/n} \cdots (b_K)^{n_K/n} \right]^n \\ &= \left[ 2^{\frac{n_1}{n} \log_2 b_1} \cdots 2^{\frac{n_K}{n} \log_2 b_K} \right]^n = \left[ 2^{\sum_{i=1}^K \frac{n_i}{n} \log_2 b_i} \right]^n. \end{aligned} \quad (4.38)$$

Then using (4.37),

$$y_1 \cdots y_n \approx \left[ 2^{\sum_{i=1}^K p_Y(b_i) \log_2 b_i} \right]^n = \left[ 2^{E[\log_2 Y]} \right]^n, \quad (4.39)$$

with high probability. A rigorous proof is left to Problem 4-16. Since (4.39) is true for any discrete-valued random variable  $Y$ , it is certainly true for a random variable

$$Y = f(X), \quad (4.40)$$

where  $f$  is any function defined on the alphabet of  $X$ . Define  $f(x) = p_X(x) = \Pr[X = x]$  for all  $x \in \Omega_X$ , certainly a legitimate function defined on the alphabet of  $X$ . Then (4.39) implies that for large  $n$

$$\begin{aligned} y_1 \cdots y_n &= f(x_1) \cdots f(x_n) = \prod_{i=1}^n f(x_i) = \prod_{i=1}^n p_X(x_i) \\ &\approx \left[ 2^{E[\log_2 p_X(X)]} \right]^n = 2^{-nH(X)} \end{aligned} \quad (4.41)$$

with high probability. Since the  $X_i$  are independent,

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^n p_X(x_i) \quad (4.42)$$

so with high probability (4.6) holds.

## PROBLEMS

- 4-1. Consider an unfair coin that produces heads with probability  $1/4$ . What is the entropy of the coin flip outcome? Suppose the coin is flipped once per second. What is the rate in this source? Devise a coder to encode successive coin flips outcomes so that the average number of bits per flip is less than one. How does your coder compare with the rate of the source?
- 4-2. Consider a random variable  $X$  with alphabet  $\Omega_X = \{a_1, a_2, a_3, a_4\}$  and probabilities

$$p_X(a_1) = 1/2 \quad p_X(a_2) = 1/4 \quad p_X(a_3) = 1/8 \quad p_X(a_4) = 1/8. \quad (4.43)$$

Find the entropy of the random variable. Suppose independent trials of the random variable

occur at rate  $r = 100$  trials/second. What is the rate of the source? Devise a coder that exactly achieves the rate of the source.

- 4-3. The well known *Jensen's inequality* from probability theory implies that

$$E[\log_2 X] \leq \log_2 E[X].$$

Use this to prove the *p-q inequality*: Given  $p_i$  and  $q_i$ , both strictly positive and defined for  $i \in \{1, 2, \dots, M\}$  such that

$$\sum_{i=1}^M p_i = 1$$

(so  $p_i$  could be a probability distribution) and

$$\sum_{i=1}^M q_i = \alpha > 0$$

then

$$-\sum_{i=1}^M p_i \log_2 p_i \leq -\sum_{i=1}^M p_i \log_2 q_i + \log_2 \alpha$$

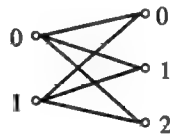
with equality if and only if  $q_i = \alpha p_i$  for all  $i$ .

- 4-4. For a discrete-valued random variable  $X$ , use the p-q inequality of Problem 4-3 to give another derivation of the results in Exercise 4-1.
- 4-5. Let  $\mathbf{X}$  denote a vector of  $n$  i.i.d. random variables each taking the value zero or one. Show that

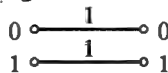
$$H(\mathbf{X}) \leq n \quad (4.44)$$

with equality if and only if the two outcomes have equal probability.

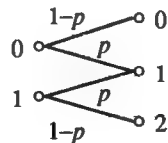
- 4-6. Consider the following discrete memoryless channel, where all transition probabilities are 1/3:



- (a) Find the two conditional entropies and the mutual information in terms of the input and output entropies.
- (b) Find the channel capacity.
- 4-7. Repeat Problem 4-6 for the following channel:



- 4-8. Repeat Problem 4-6 for the following channel, called a binary erasure channel:



(Answer to b:  $C_s = 1 - p$ .)

- 4-9.

- (a) Show that when  $p_i$  and  $q_i$  are probability distributions,

$$-\sum_i p_i \log_2 p_i \leq -\sum_i p_i \log_2 q_i. \quad (4.45)$$

- (b) Use (4.45) to establish the result of Exercise 4-1.
- 4-10. Consider a cascade of  $L$  BSC's each with the same transition probability, where the output of each BSC is connected to the input of the next.
- (a) Show that the resulting overall channel is a BSC.
- (b) Find the error probability of the overall channel as a function of  $L$ .
- (c) What happens as  $L \rightarrow \infty$ ?
- 4-11. Consider a distribution  $\{p_i, 1 \leq i \leq K\}$ , where  $p_1 > p_2$ . Further define a second distribution  $\{q_i, 1 \leq i \leq K\}$ , where  $q_1 = p_1 - \delta$  and  $q_2 = p_2 + \delta$  and  $q_i = p_i, i > 2$ , where  $\delta > 0$ . Show that the second distribution has larger entropy. **Hint:** Use the results of Problem 4-9.
- 4-12. Consider a continuous-valued random variable  $X$  uniformly distributed on the interval  $[-a, a]$ :
- (a) What is its entropy?
- (b) How does its entropy compare to that of a Gaussian distribution with the same variance?
- 4-13. Use the p-q inequality of Problem 4-3 to show the following.
- (a) For any two discrete-valued random variables  $X$  and  $Y, I(X, Y) \geq 0$ .
- (b)  $H(X) \geq H(X|Y)$
- (c)  $H(X) + H(Y) \geq H(X, Y)$
- (d) When are these inequalities equalities?
- 4-14. Show that by replacing the summations in (4.24) with integrals, the mutual information of two continuous-valued random variables can be written

$$I(X, Y) = \int_{\Omega_X} \int_{\Omega_Y} f_{X,Y}(x, y) \log_2 \frac{f_{X,Y}(x, y)}{f_X(x)f_Y(y)} dy dx. \quad (4.46)$$

- 4-15. Investigate the capacity of the vector Gaussian channel of (4.36) as the number of degrees of freedom  $N$  increases. Interpret the result.
- 4-16. Consider a random process  $\{X_k\}$ , where the components are independent observations of a random variable  $X$ . The law of large numbers for sums of random variables states that for any  $\epsilon > 0$ ,

$$\Pr \left[ E[X] - \epsilon < \frac{1}{n}(X_1 + \cdots + X_n) < E[X] + \epsilon \right] \rightarrow 1 \text{ as } n \rightarrow \infty. \quad (4.47)$$

Use this to prove (4.39).

- 4-17. Consider the following betting game. You bet \$100 and toss a die. If a six comes up, you win \$500, otherwise you win \$20. Next you bet your \$500 or \$20 and the game is repeated with the same rate of return. In other words, on the  $n^{\text{th}}$  iteration, you bet  $M_n$  dollars (your previous winnings) and win  $5M_n$  if a six comes up and  $M_n/5$  otherwise. What is the expected value of the money you have after  $n$  flips? Show that with high probability,  $M_n$  goes to zero for large  $n$ . Would you play this game?
- 4-18. Consider an analog continuous-time communication circuit with cascaded amplifiers. Suppose that the amplifiers have random gain, each independently taken from the same distribution. If the number of amplifiers is large, which is a better estimate of the gain of the system, (a) the expected value of the product of the gains of the amplifiers, or (b) the expected value of the sum of the gains expressed in dB?

## REFERENCES

1. T. Berger, *Rate Distortion Theory*, Prentice-Hall, Englewood Cliffs, NJ (1971).
2. G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Information Theory* IT-28, No. 1(Jan. 1982).
3. N. Abramson, *Information Theory and Coding*, McGraw-Hill Book Co., New York (1963).
4. R. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, Inc., New York (1968).
5. R. J. McEliece, *The Theory of Information and Coding*, Addison Wesley Pub. Co. (1977).
6. T. M. Cover and J. A. Thomas, "Elements of Information Theory," Wiley, (1991).
7. R. E. Blahut, "Principles and Practice of Information Theory," Addison-Wesley, (1987).
8. D. Slepian (editor), *Key Papers in the Development of Information Theory*, IEEE Press, New York (1974).
9. C. E. Shannon, "A Mathematical Theory of Communication," *BSTJ*, (Oct. 1948).
10. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Illinois (1963).
11. A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill (1979).
12. J. Wolfowitz, *The Coding Theorems of Information Theory*, 3d ed., Springer-Verlag, Berlin (1978).

# 5

---

## PHYSICAL MEDIA AND CHANNELS

---

Ultimately the design of a digital communication system depends on the properties of the channel. The channel is typically a part of the digital communication system that we cannot change. Some channels are simply a physical medium, such as a wire pair or optical fiber. On the other hand, the radio channel is part of the electromagnetic spectrum, which is divided by government regulatory bodies into bandlimited radio channels that occupy disjoint frequency bands. In this book we do not consider the design of the *transducers*, such as antennas, lasers, and photodetectors, and hence we consider them part of the channel. Some channels, notably the telephone channel, are actually composites of multiple transmission subsystems. Such *composite channels* derive their characteristics from the properties of the underlying subsystems.

Section 5.1 discusses composite channels. Sections 5.2 through 5.4 review the characteristics of the most common channels used for digital communication, including the transmission line (wire pair or coaxial cable), optical fiber, and microwave radio (satellite, point-to-point and mobile terrestrial radio). Section 5.5 discusses the composite voiceband telephone channel, which is often used for voiceband data transmission. Finally, Section 5.6 discusses magnetic recording of digital data, as used in tape and disk drives, which has characteristics similar in many ways to the other channels discussed.

The most prevalent media for new installations in the future will be optical fiber and microwave radio, and possibly lossless transmission lines based on

superconducting materials. However, there is a continuing strong interest in lossy transmission lines and voiceband channels because of their prevalence in existing installations. Thus all the media discussed in this chapter are important in new applications of digital communication.

## 5.1. COMPOSITE CHANNELS

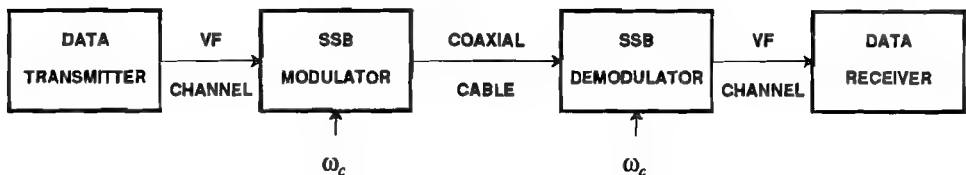
It is common for many users to share a common communication medium, for example by *time-division* and *frequency-division multiplexing* (Chapter 16).

### Example 5-1.

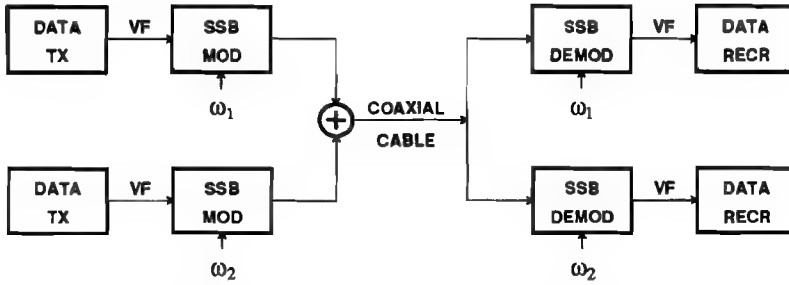
Voice signals are roughly bandlimited to frequencies lower than 4 kHz. A suitable baseband channel therefore needs to pass only frequencies up to 4 kHz. Such a channel is often derived from a much higher bandwidth physical medium that is shared with other users. A *voice frequency (VF)* channel derived from a coaxial cable (Section 5.2) using *single-sideband modulation* is shown in Figure 5-1. The SSB modulator translates the VF channel to the neighborhood of a frequency  $\omega_c$  for transmission on the coaxial cable. A VF channel can be used for digital communication, as long as the modulation technique conforms to the limitations of the channel.  $\square$

The channel in Example 5-1 is an example of a *composite channel*, because it consists of multiple subsystems. If the VF channel was designed for voice transmission, it has certain characteristics which are beyond the control of the designer of the digital communication system. The VF channel characteristics in this case depend not only on the properties of the physical medium, but also on the design of the SSB modulation system.

Composite channels usually arise in the context of *multiple access*, which is defined as access to a physical medium by two or more independent users. This is again illustrated by example.



**Figure 5-1.** Composite data channel derived from an SSB modulation system, where  $\omega_c$  is the carrier frequency.



**Figure 5-2.** Two data channels derived from a single coaxial cable by FDM, where  $\omega_1$  and  $\omega_2$  are distinct carrier frequencies. "TX" is the transmitter and "RECR" is the receiver.

**Example 5-2.**

Figure 5-2 shows a *frequency-division multiplexing (FDM)* approach using SSB modulation. In this case two VF channels are derived from a single coaxial cable using SSB modulation. The two channels are separated by using two different carrier frequencies in the two modulators,  $\omega_1$  and  $\omega_2$ , where these frequencies are chosen far enough apart that the spectra for the two modulated VF channels do not overlap. These two VF channels can be used independently for digital communication. □

In fact FDM is a very common technique in the telephone network for deriving many VF channels from a single physical medium such as coaxial cable or microwave radio (thousands rather than just two as in the example!).

Another common composite channel is illustrated in the following example.

**Example 5-3.**

A VF channel derived from a digital transmission system using *pulse-code modulation (PCM)* is illustrated in Example 5-3. The PCM system samples the VF channel at 8 kHz, corresponding to a maximum bandwidth of 4 kHz, and then quantizes each sample to eight bits. The total bit rate for the PCM encoded VF channel is 64 kb/s. This derived VF channel may be used for data transmission; again, any digital modulation technique can be used subject to basic constraints imposed by the PCM system. The total bit rate that can be transmitted through this derived VF channel is less than 64 kb/s, in fact more of the order of 20-30 kb/s. The direct transmission of the bit stream over the digital transmission system would obviously be more efficient, but the situation in Figure 5-3 is still very common due



**Figure 5-3.** Data transmission over a PCM-derived VF channel.



to the presence of much existing PCM equipment for voice transmission and the desire to transmit data over a channel designed primarily for voice. □

Physical media as well as the composite channels derived from them impose constraints on the design of a digital communication system. Many of these constraints will be mentioned in this chapter for particular media. The nature of these constraints usually fall within some broad categories:

- A *bandwidth constraint*. Sometimes this is in the form of a channel attenuation which increases gradually at high frequencies, and sometimes (particularly in the case of composite channels) it is in the form of a very hard bandwidth limit.
- A *transmitted power constraint*. This is often imposed to limit interference of one digital communication system with another, or imposed by the inability of a composite channel to transmit a power level greater than some threshold, or by a limitation imposed by the power supply voltage of the digital communication system itself. This power constraint can be in the form of a *peak-power constraint*, which is essentially a limit on the transmitted voltage, or can be an *average power constraint*.

#### Example 5-4.

An FDM system such as that in Figure 5-2, where there are perhaps thousands of channels multiplexed together, is designed under assumptions on the average power of the channels. From this average power, the total power of the multiplexed signal can be deduced; this power is adjusted relative to the point at which amplifiers in the system start to become nonlinear. If a significant number of VF channels violate the average power constraint, then the multiplexed signal will overload the amplifiers, and the resulting nonlinearity will cause *intermodulation distortion* and interference between VF channels. □

#### Example 5-5.

The PCM system of Figure 5-3 imposes a bandwidth constraint on the data signal, which must be less than half the sampling rate. A peak power constraint is also imposed by the fact that the quantizer in the PCM system has an overload point beyond which it clips the input signal. □

#### Example 5-6.

The regenerative repeaters in Figure 1-3 are usually powered by placing a high voltage on the end of the system and then stringing the repeaters in series (like Christmas tree lights!); a fraction of the total voltage appears across each repeater. The power consumption of the repeaters is limited by the applied voltage, the ohmic loss of the cable, and the number of repeaters. This places an average power constraint on the transmitted signal at each repeater. In practice there is also a peak power constraint due to the desire not to have to generate a signal voltage higher than the supply voltage drop across the repeater. An additional factor limiting the transmitted power for wire-pair systems is crosstalk into other communication systems in the same multi-pair cable, as discussed in Section 5.2.4. □

**Example 5-7.**

The optical fiber medium (Section 5.3) becomes significantly nonlinear when the input power exceeds about one milliwatt. Thus, in many applications there is a practical limit on the average transmitted power.  $\square$

In addition to placing constraints on the transmitted signal, the medium or composite channel introduces *impairments* which limit the rate at which we can communicate. We will see many examples of this in this Chapter.

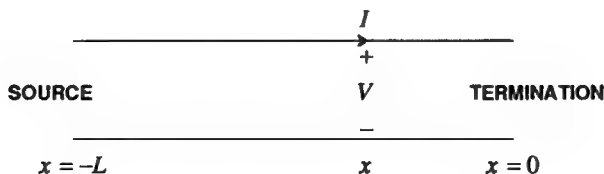
## 5.2. TRANSMISSION LINES

One of the most common media for data transmission in the past has been the transmission line composed of a pair of wires or a coaxial cable. Coaxial cable is commonly used for digital communication within a building, as in a local-area network, and for high-capacity long-distance facilities in the telephone network. Wire pairs are much more extensively used, primarily for relatively short distance trunking in the telephone network between switching machines in metropolitan areas. The spacing between regenerative repeaters is typically about 1.5 km, with bit rates in the range of 1.5-6 Mb/s on wire pair to 270-400 Mb/s on coaxial cable. In addition, wire pairs are used for connection of the telephone instrument to the central office, and while this connection is primarily for analog voiceband transmission, there is work proceeding to use this same medium for digital communication at 144 kb/s or higher in the Integrated Services Digital Network (ISDN). This is called the *digital subscriber loop*, and requires a distance for transmission of about 4-5 kilometers without repeaters.

### 5.2.1. Review of Transmission Line Theory

A *uniform transmission line* is a two-conductor cable with a uniform cross-section. It may consist of a pair of wires twisted together (*twisted wire cable*) or a cable with a cylindrical outer conductor surrounding a wire (*coaxial cable*). While the details of the cable characteristics depend on the cross-section geometry, the basic theory does not.

A uniform transmission line can be represented by a pair of conductors as shown



**Figure 5-4.** A uniform transmission line, where  $x$  is the distance along the line.

in Figure 5-4. We denote the termination of the line on the right as  $x = 0$ , and the source on the left as  $x = -L$ , where  $x$  is the distance along the line and  $L$  is the length of the line. Assume that the line is excited with a complex exponential with radian frequency  $\omega$ . The voltage and current along the line will be a function of both the frequency  $\omega$  and the distance  $x$ . Writing the voltage and current at a point  $x$ , the dependence on time is given by the complex exponential,

$$V(x, \omega) = V(x)e^{j\omega t}, \quad I(x, \omega) = I(x)e^{j\omega t} \quad (5.1)$$

where  $V(x)$  and  $I(x)$  are complex numbers which summarize the amplitude and phase of the complex exponential at distance  $x$ .

The voltage and current as a function of distance along the line consists of two propagating waves, one from source to termination and the other from termination to source. The first we call the *source wave*, and the latter we call the *reflected wave*. The total voltages and currents are the sum of the two waves, given by

$$V(x) = V_+e^{-\gamma x} + V_-e^{\gamma x}, \quad I(x) = \frac{1}{Z_0}(V_+e^{-\gamma x} - V_-e^{\gamma x}) \quad (5.2)$$

In these equations the  $V_+$  terms correspond to the source wave, and the  $V_-$  terms correspond to the reflected wave. The complex impedance  $Z_0$  is called the *characteristic impedance* of the transmission line, since it equals the ratio of the voltage to current at any point of the line (independent of  $x$ ) for either the source or reflected wave. The other complex quantity in this equation is  $\gamma$ , which is called the *propagation constant*. The real and imaginary parts of  $\gamma$  are of importance in their own right, so in

$$\gamma = \alpha + j\beta \quad (5.3)$$

the real part  $\alpha$  and imaginary part  $\beta$  are called respectively the *attenuation constant* and *phase constant*. The attenuation constant has the units of *neper per unit distance* and the phase constant has the units of *radians per unit distance*.

There are three things that distinguish the source wave and its reflection:

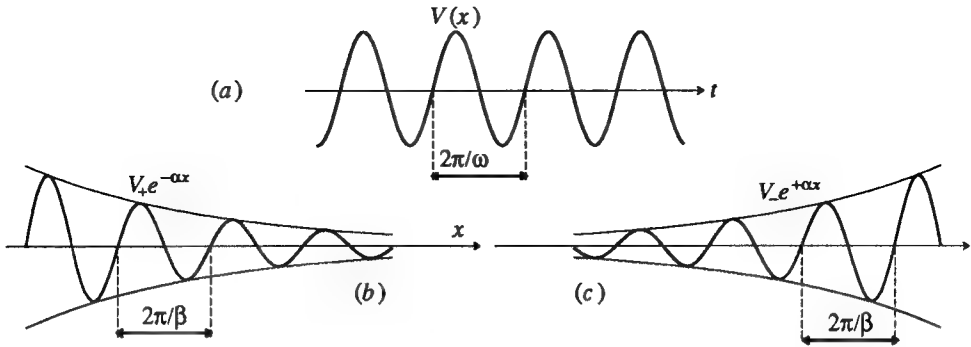
- The amplitude and phase as expressed by  $V_+$  and  $V_-$  are different.
- The current is flowing in opposite directions.
- The sign of the exponent is different.

The third difference is illustrated in Figure 5-5. The dependence of both waves on time at any point along the line is shown in Figure 5-5a. This is of course just a sinusoid of constant amplitude and phase, where both amplitude and phase depend on the frequency of the wave. For a fixed time  $t = t_0$ , the source wave is shown in Figure 5-5b as a function of distance  $x$ , and is given by

$$V(x) = V_+e^{-\alpha x}e^{-j\beta x}. \quad (5.4)$$

The amplitude of this wave decreases with distance  $x$ . The *wavelength* of the wave, or distance between nulls, is  $2\pi/\beta$ .

The phase shift of the complex exponential with radian frequency  $\omega$  for a transmission line of length  $L$  is  $\beta L$  radians, which corresponds to  $\beta L/2\pi$  cycles. This



**Figure 5-5.** The voltage on a uniform transmission line. a. The voltage at one point in the line as a function of time. b. The magnitude of the voltage vs. distance for the source wave at a fixed time. c. (b) repeated for the reflected wave.

phase shift represents a propagation delay of the sinusoid, and we can readily figure out the size of the delay. Since each cycle corresponds to  $2\pi/\omega$  seconds from Figure 5-5a, it follows that the total delay of the complex exponential is

$$\frac{\beta L}{2\pi} \text{ cycles} \cdot \frac{2\pi}{\omega} \frac{\text{sec}}{\text{cycles}} = \frac{\beta}{\omega} L \text{ sec.} \quad (5.5)$$

The propagation velocity of the wave on the transmission line is therefore related to the frequency and phase constant by

$$v = \frac{\omega}{\beta}. \quad (5.6)$$

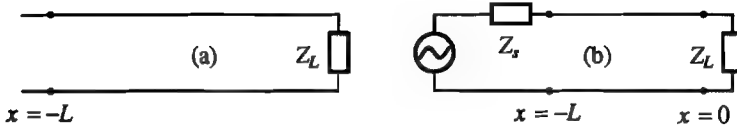
Since  $\alpha$  is always greater than zero, the magnitude of the wave is also decaying exponentially with distance in accordance with the term  $e^{-\alpha x}$ . This implies that at any frequency the loss of the line in dB is proportional to the length of the line. We get a power loss in dB

$$\gamma_0 L = 10 \log_{10} \left( \frac{P_T}{P_R} \right), \quad (5.7)$$

where  $\gamma_0 = 20 \alpha \log_{10} e$  is the loss in dB per unit distance. Since  $\alpha$  is frequency dependent, so to is  $\gamma_0$ .

Similarly, shown in Figure 5-5c is the reflected wave amplitude as a function of distance along the line. This wave is also decaying exponentially with distance in the direction of propagation, which is from termination to source.

Based on these relationships, we can determine the voltage along the line for any source and termination impedances by simply matching up boundary conditions.



**Figure 5-6.** A terminated transmission line. a. Without a source termination. b. With a source termination.

### Example 5-8.

A transmission line terminated in impedance  $Z_L$  is shown in Figure 5-6a. What is the relative size of the incident and reflected waves at the termination? This quantity is called the *voltage reflection coefficient* and is usually denoted by  $\Gamma$ . The boundary condition is that  $Z_L$  is the ratio of the voltage to current at  $x = 0$ , so from (5.2),

$$Z_L = Z_0 \frac{V_+ + V_-}{V_+ - V_-}, \quad \Gamma = \frac{V_-}{V_+} = \frac{Z_L - Z_0}{Z_L + Z_0}. \quad (5.8)$$

Several special cases are of interest. When the load impedance is equal to the characteristic impedance,  $Z_L = Z_0$ , then the reflection coefficient is zero,  $\Gamma = 0$ . When the line is open circuited,  $Z_L = \infty$ , then  $\Gamma = 1$  indicating that the reflected voltage is the same as the incident wave at the point of the open circuit. Finally, when the line is closed circuited,  $Z_L = 0$ , then  $\Gamma = -1$  indicating that the reflected voltage is the negative of the incident voltage.  $\square$

### Example 5-9.

For the terminated transmission line of Figure 5-6a, what is the impedance looking into the line as a function of its length? Taking the ratio of the voltage to current from (5.2) and (5.8),

$$\frac{V(x)}{I(x)} = Z_0 \frac{e^{-\gamma x} + \Gamma e^{\gamma x}}{e^{-\gamma x} - \Gamma e^{\gamma x}}, \quad Z_{in} = \frac{V(-L)}{I(-L)} = Z_0 \frac{1 + \Gamma e^{-2\gamma L}}{1 - \Gamma e^{-2\gamma L}}. \quad (5.9)$$

When the line is terminated in its characteristic impedance,  $\Gamma = 0$  and the input impedance is equal to the characteristic impedance.  $\square$

### Example 5-10.

For the terminated line of Figure 5-6b with a source impedance of  $Z_s$ , what is the voltage transfer function from source  $V_{in}$  to the load? Writing the node voltage equation at the source,

$$V(-L) = V_{in} - I(-L)Z_s \quad (5.10)$$

and we have the two additional relations

$$V(-L) = V_+(e^{\gamma L} + \Gamma e^{-\gamma L}), \quad I(-L) = \frac{V_+}{Z_0}(e^{\gamma L} - \Gamma e^{-\gamma L}), \quad (5.11)$$

which enable us to solve for the three constants  $V_+$ ,  $V(-L)$ , and  $I(-L)$ . Finally, the output voltage is

$$V(0) = V_+(1 + \Gamma) \quad (5.12)$$

Putting this all together, the desired voltage transfer function is

$$\frac{V(0)}{V_{in}} = \frac{Z_0(1 + \Gamma)}{(Z_0 + Z_S)e^{\gamma L} + \Gamma(Z_0 - Z_S)e^{-\gamma L}} \quad (5.13)$$

When the source impedance is equal to the characteristic impedance,  $Z_S = Z_0$  this simplifies to

$$\frac{V(0)}{V_{in}} = \frac{1 + \Gamma}{2} e^{-\gamma L}. \quad (5.14)$$

When further the line is terminated in its characteristic impedance,  $\Gamma = 0$  and the transfer function consists of the attenuation and phase shift of the transmission line (times another factor of 0.5 corresponding to the attenuation due to the source and termination impedances). When the line is short-circuited, then the transfer function is zero as expected since  $\Gamma = -1$ . What happens when the line is open circuited?  $\square$

Transmission lines are often analyzed, particularly where computer programs are written, using the concept of a *chain matrix* [1]. A *twoport network* is a network as illustrated in Figure 5-7, which has an input port and output port and for which the input and output currents are complementary as shown. The chain matrix relates the input voltage and current to the output voltage and current, viz.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \quad (5.15)$$

where all quantities are complex functions of frequency. The chain matrix characterizes the twoport transfer function completely, and can be used to analyze connections of twoports (such as transmission lines, thereby serving to analyze nonuniform transmission lines). Its importance arises from the following fact.

#### Exercise 5-1.

Show that if two twoports are connected in series, then the chain matrix of the combination twoport is the product of the first chain matrix times the second chain matrix.  $\square$

The chain matrix of a uniform transmission line is easily calculated, giving us a ready technique for systematically analyzing combinations of transmission lines with other



**Figure 5-7.** Illustration of a twoport network with definition of voltages and currents for the chain matrix.

circuit elements.

### Exercise 5-2.

Show that the chain matrix of a uniform transmission line is

$$\begin{bmatrix} \cosh(\gamma L) & Z_0 \sinh(\gamma L) \\ \frac{\sinh(\gamma L)}{Z_0} & \cosh(\gamma L) \end{bmatrix}. \quad (5.16)$$

□

### 5.2.2. Cable Primary Constants

The characteristic impedance and propagation constant are called *secondary parameters* of the cable because they are not related directly to physical parameters. A simple model for a short section of the transmission line is shown in Figure 5-8. This model is in terms of four parameters: the conductance  $G$  in mhos per unit length, the capacitance  $C$  in farads per unit length, the inductance  $L$  in henries per unit length, and the resistance  $R$  in ohms per unit length. All of these parameters of the transmission line are functions of frequency in general, and they differ for different cross-sections (for example, twisted pair vs. coaxial cable). In general these parameters are determined experimentally for a given cable.

This lumped-parameter model becomes an exact model for the transmission line as the length of the line  $dx \rightarrow 0$ , and is useful since it displays directly physically meaningful quantities. These parameters are called the *primary constants* of the transmission line. The secondary parameters can be calculated directly in terms of the primary constants as

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}, \quad \gamma = \sqrt{(R + j\omega L)(G + j\omega C)}. \quad (5.17)$$

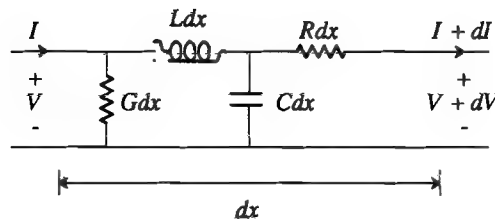


Figure 5-8. Lumped-parameter model for a short section of transmission line.

**Example 5-11.**

A *lossless transmission* line is missing the two dissipative elements, resistance and conductance. New superconducting materials show promise of actually being able to realize this ideal. The secondary parameters in this case are

$$Z_0 = \sqrt{\frac{L}{C}}, \quad \gamma = j\omega\sqrt{LC}. \quad (5.18)$$

The characteristic impedance of a lossless transmission line is real-valued and hence resistive. A pure resistive termination is often used as a reasonable approximation to the characteristic impedance of a *lossy* transmission line, although the actual characteristic impedance increases at low frequencies and includes a capacitive reactive component. The propagation constant is imaginary, and since  $\alpha = 0$  the lossless transmission line has no attenuation as expected. The propagation velocity on a lossless transmission line is

$$v = \frac{\omega}{\beta} = \frac{1}{\sqrt{LC}}. \quad (5.19)$$

□

The primary constants of actual cables depend on many factors such as the geometry and the material used in the insulation. For twisted wire pairs [2] the capacitance is independent of frequency for the range of frequencies of interest (0.083  $\mu$ Farads per mile, or 0.0515  $\mu$ Farads per kilometer is typical), the conductance is negligibly small, the inductance is a slowly varying function of frequency decreasing from about one mili-Henries (mH) per mile or 0.62 mH per km, at low frequencies to about 70% of that value at high frequencies, and the resistance is proportional to the square root of frequencies at high frequencies due to the *skin effect* (the tendency of the current to flow near the surface of the conductor, increasing the resistance).

**Example 5-12.**

What is the velocity of propagation on a twisted wire cable? From (5.19), for a lossless line

$$v = \frac{1}{\sqrt{(0.083 \cdot 10^{-6})(10^{-3})}} = 1.1 \cdot 10^5 \text{ miles/sec} = 1.76 \cdot 10^5 \text{ km/sec}. \quad (5.20)$$

Since the speed of light in freespace is  $3 \cdot 10^5$  km/sec, the velocity on the line is a little greater than half the speed of light. The delay is about 5.65  $\mu$ sec per km. This approximation is valid on practical twisted wire pairs for frequencies where  $R \ll \omega L$ . □

Coaxial cable is popular for higher frequency applications primarily because the outer conductor effectively shields against radiation to the outside world and conversely interference from outside sources. At lower frequencies near voiceband this shielding is ineffective and hence the coaxial cable does not have any advantage over the more economical twisted wire pair. In terms of primary constants, the main difference between coaxial cable and wire pair is that the coaxial inductance is essentially independent of frequency.

**Example 5-13.**

A more accurate model than Example 5-11 of a cable, wire pair or coaxial, would be to assume that  $G$  only is zero. Then the propagation constant of (5.17) becomes



$$\alpha = \omega \sqrt{\frac{LC}{2}} \left\{ \left( 1 + \frac{R^2}{\omega^2 L^2} \right)^{1/2} - 1 \right\}^{1/2}, \quad \beta = \omega \sqrt{\frac{LC}{2}} \left\{ \left( 1 + \frac{R^2}{\omega^2 L^2} \right)^{1/2} + 1 \right\}^{1/2}. \quad (5.21)$$

At frequencies where  $R \ll \omega L$ ,

$$\alpha \approx \frac{R}{2} \sqrt{\frac{C}{L}} \text{ nepers per unit length} \quad (5.22)$$

$$\beta \approx \omega \sqrt{LC}. \quad (5.23)$$

Hence the velocity relation of (5.19) is still valid in this range of frequencies. Since at high frequencies  $R$  increases as the square root of frequency, the attenuation constant in nepers (or dB) has the same dependency. It follows that the loss of the line in dB at high frequencies is proportional to the square root of frequency.  $\square$

#### Example 5-14.

If the loss of a cable is 40 dB at 1 Mhz, what is the approximate loss at 4 MHz? The answer is 40 dB times the square root of 4, or 80 dB.  $\square$

The results of Example 5-13 suggest that the propagation constant is proportional to frequency. This *linear phase* model suggests that the line offers, in addition to attenuation, a constant delay at all frequencies. However, a more refined model of the propagation constant [3] shows that there is an additional term in the phase constant proportional to the square root of frequency. This implies that the cable will have some *group delay*, which is delay dependent on frequency. This implies that the different frequency components of a pulse launched into the line will arrive at the termination with slightly different delays. Both the frequency-dependent attenuation and the group delay cause *dispersion* on the transmission line, or spreading in time of a transmitted pulse. The attenuation causes dispersion because the bandlimiting effect broadens the pulse, and delay distortion causes dispersion because the different frequency components arrive with different delays.

### 5.2.3. Impedance Discontinuities

The theory presented thus far has considered a single uniform transmission line. In practice, it is common to encounter different *gauges* (diameters) of wire connected together. These gauge changes do not affect the transmission materially, except for introducing a slight discontinuity in impedance, which will result in small reflections. A more serious problem for digital transmission in the subscriber loop between central office and customer premises is the *bridged tap*, an additional open circuited wire pair bridged onto the main cable pair.

### 5.2.4. Crosstalk

An important consideration in the design of a digital communication system using a transmission line as a physical medium is the *range* or *distance* which can be achieved between regenerative repeaters. This range is generally limited by the high frequency gain which must be inserted into the receiver equalization to compensate for cable attenuation. This gain amplifies noise and interference signals which may be present, causing the signal to deteriorate as the range increases. The most important

noise and interference signals are thermal noise (due to random motion of electrons), impulse noise (caused by switching relays and similar mechanisms), and crosstalk between cable pairs. Crosstalk, and interference from external sources such as power lines, can be minimized by using *balanced transmission*, in which the signal is transmitted and received as a difference in voltage between the two wires; this helps because external interference couples approximately equally into the two wires and hence is approximately canceled when the difference in voltage is taken at the receiver. A common way to achieve balanced transmission is to use *transformer coupling* of the transmitter and receiver to the wire pair; in addition, this affords additional protection against damage to the electronics due to foreign potentials such as lightning strikes.

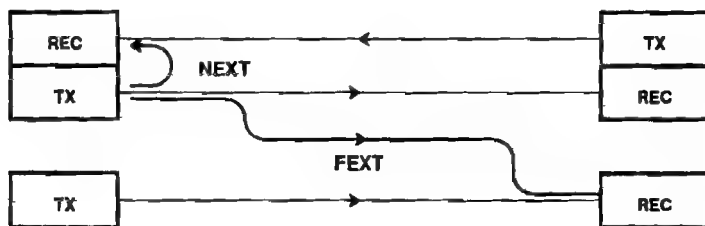
There are two basic crosstalk mechanisms, *near-end crosstalk (NEXT)* and *far-end crosstalk (FEXT)*, illustrated in Figure 5-9. NEXT [4] represents a crosstalk of a local transmitter into a local receiver, and experiences an attenuation which is accurately modeled by

$$|H_{\text{NEXT}}(j\omega)|^2 = K_{\text{NEXT}} |\omega|^{1.5} \quad (5.24)$$

where  $H_{\text{NEXT}}(j\omega)$  is the transfer function experienced by the crosstalk. FEXT represents a crosstalk of a local transmitter into a remote receiver, with an attenuation given by

$$|H_{\text{FEXT}}(j\omega)|^2 = K_{\text{FEXT}} |C(j\omega)|^2 |\omega|^2 \quad (5.25)$$

where  $C(f)$  is the loss of the cable. Where present, NEXT will dominate FEXT because FEXT experiences the loss of the full length of the cable (in addition to the crosstalk coupling loss) and NEXT does not. Both forms of crosstalk experience less attenuation as frequency increases, and hence it is advantageous to minimize the bandwidth required for transmission in a crosstalk limited environment.



**Figure 5-9.** Illustration of two types of crosstalk -- far-end crosstalk (FEXT) and near-end crosstalk (NEXT).

### 5.3. OPTICAL FIBER

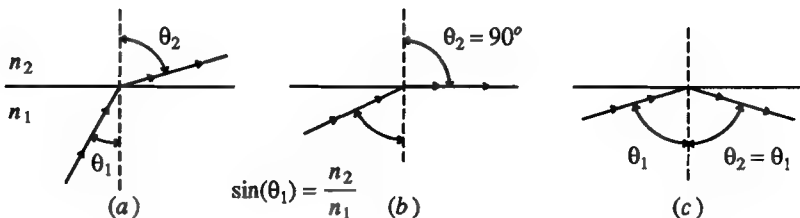
The optical fiber cable is capable of transmitting light for long distances with high bandwidth and low attenuation. Not only this, but it offers freedom from external interference, immunity from interception by external means, and inexpensive and abundant raw materials. It is difficult to imagine a more ideal medium for digital communication!

The use of an optical dielectric waveguide for high performance communication was first suggested by Kao and Hockham in 1966 [5]. By 1986 this medium was well developed and was rapidly replacing wire pairs and coax in many new cable installations. It allows such a great bandwidth at modest cost that it will also replace many of the present uses for satellite and radio transmission. Thus, it appears that digital communication over wire-pairs and coax will be mostly limited to applications constrained to use existing transmission facilities (such as in the digital subscriber loop).

Digital transmission by satellite and radio will be limited to special applications that can make use of their special properties. For example, radio communication is indispensable for situations where one or both terminals are mobile, for example in digital mobile telephony (Section 5.4) or deep space communication. Radio is also excellent for easily bridging geographical obstacles such as rivers and mountains, and satellite is excellent for spanning long distances where the total required bandwidth is modest and the installation of an optical fiber cable would not be justified. Furthermore, satellite has unique capabilities for certain types of multiple-access situations (Chapter 18) spread over a wide geographical area.

#### 5.3.1. Fiber Optic Waveguide

The principle of an optical fiber waveguide [6] can be understood from the concept of *total internal reflection*, shown in Figure 5-10. A light wave, represented by a single ray, is incident on a boundary between two materials, where the angle of incidence is  $\theta_1$  and the angle of refraction is  $\theta_2$ . We define a *ray* as the path that the center of a slowly diverging beam of light takes as it passes through the system; such



**Figure 5-10.** Illustration of Snell's Law and total internal reflection. a. Definition of angles of incidence  $\theta_1$  and refraction  $\theta_2$ . The angle of refraction is larger if  $n_1 > n_2$ . b. The critical angle of incidence at which the angle of refraction is ninety degrees. c. At angles larger than the critical angle, total internal reflection occurs.

a beam must have a diameter large with respect to the wavelength in order to be approximated as a plane wave [7]. Assuming that the index of refraction  $n_1$  in the incident material is greater than the index of refraction of the refractive medium,  $n_2$ , or  $n_1 > n_2$ . Then Snell's Law predicts that

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{n_2}{n_1} < 1. \quad (5.26)$$

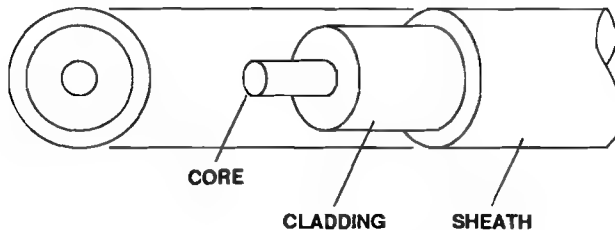
The angle of refraction is larger than the angle of incidence. Shown in Figure 5-10b is the case of a critical incidence angle where the angle of refraction is ninety degrees, so that the light is refracted along the material interface. This corresponds to *critical* incident angle

$$\sin(\theta_1) = \frac{n_2}{n_1}. \quad (5.27)$$

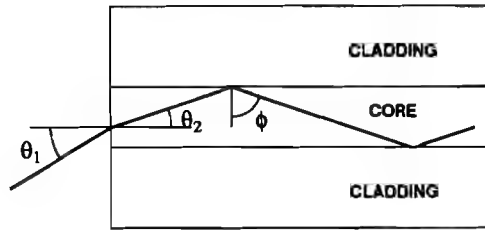
For angles larger than (5.27), there is total internal reflection as illustrated in Figure 5-10c, where the angle of reflection is always equal to the angle of incidence.

This principle can be exploited in an *optical fiber waveguide* as illustrated in Figure 5-11. The *core* and *cladding* materials are glass, which transmits light with little attenuation, while the *sheath* is an opaque plastic material that serves no purpose other than to lend strength, absorb any light that might otherwise escape, and prevent any light from entering (which would represent interference or crosstalk). The core glass has a higher index of refraction than the cladding, with the result that incident rays with a small angle of incidence are captured by total internal reflection. This is illustrated in Figure 5-12, where a light ray incident on the end of the fiber is captured by total internal reflection as long as the angle of incidence  $\theta_1$  is below a critical angle (Problem 5-4). The ray model predicts that the light will bounce back and forth, confined to the waveguide until it emerges from the other end. Furthermore, it is obvious that the path length of a ray, and hence the transit time, is a function of the incident angle of the ray (Problem 5-5).

This variation in transit time for different rays manifests itself in *pulse broadening* — the broadening of a pulse launched into the fiber as it propagates — which in



**Figure 5-11.** An optical fiber waveguide. The core and cladding serve to confine the light incident at narrow incident angles, while the opaque sheath serves to give mechanical stability and prevent crosstalk or interference.



**Figure 5-12.** Ray model of propagation of light in an optical waveguide by total internal reflection. Shown is a cross-section of a fiber waveguide along its axis of symmetry, with an incident light ray at angle  $\theta_1$  which passes through the axis of the fiber (a meridional ray).

turn limits the pulse rate which can be used or the distance that can be transmitted or both. The pulse broadening can be reduced by modifying the design of the fiber, and specifically by using a *graded-index fiber* in which the index of refraction varies continuously with radial dimension from the axis.

The foregoing ray model gives some insight into the behavior of light in an optical fiber waveguide; for example, it correctly predicts that there is a greater pulse broadening when the index difference between core and cladding is greater. However, this model is inadequate to give an accurate description since in practice the radial dimensions of the fiber are on the order of the wavelength of the light. For example, the ray model of light predicts that there is a continuum of angles for which the light will bounce back and forth between core-cladding boundaries indefinitely. A more refined model uses Maxwell's equations to predict the behavior of light in the waveguide, and finds that in fact there are only a discrete and finite number of angles at which light propagates in zigzag fashion indefinitely. Each of these angles corresponds to a *mode* of propagation, similar to the modes in a metallic waveguide carrying microwave radiation. When the core radius is many times larger than the wavelength of the propagating light, there are many modes; this is called a *multimode* fiber. As the radius of the core is reduced, fewer and fewer modes are accommodated, until at a radius on the order of the wavelength only one mode of propagation is supported. This is called a *single mode* fiber. For a single mode fiber the ray model is seriously deficient since it depends on physical dimensions that are large relative to the wavelength for its accuracy. In fact, in the single mode fiber the light is not confined to the core, but in fact a significant fraction of the power propagates in the cladding. As the radius of the core gets smaller and smaller, more and more of the power travels in the cladding.

For various reasons, as we will see, the transmission capacity of the single mode fiber is greater. However, it is also more difficult to splice with low attenuation, and it also fails to capture light at the larger incident angles that would be captured by a multimode fiber, making it more difficult to launch a given optical power. In view of its much larger ultimate capacity, there is a trend toward exclusive use of single mode fiber in new installations, even though multimode fiber has been used extensively in the past [8]. In the following discussion, we emphasize the properties of single mode

fiber.

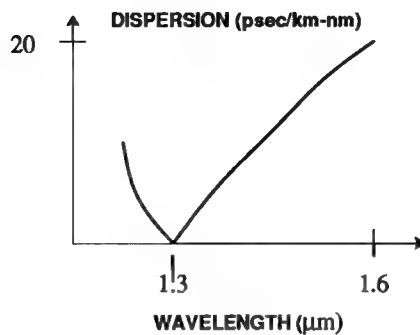
We will now discuss the factors which limit the bandwidth or bit rate which can be transmitted through a fiber of a given length. The important factors are:

- **Material attenuation**, the loss in signal power that inevitably results as light travels down an optical waveguide. There are four sources of this loss in a single mode fiber — scattering of the light by inherent inhomogeneities in the molecular structure of the glass crystal, absorption of the light by impurities in the crystal, losses in connectors, and losses introduced by bending of the fiber. Generally these losses are affected by the wavelength of the light, which affects the distribution of power between core and cladding as well as scattering and absorption mechanisms. The effect of these attenuation mechanisms is that the signal power loss in dB is proportional to the length of the fiber. Therefore, for a line of length  $L$ , if the loss in dB per kilometer is  $\gamma_0$ , the total loss of the fiber is  $\gamma_0 L$  and hence the ratio of transmitted power  $P_T$  to received power  $P_R$  obeys

$$\gamma_0 L = 10 \log_{10} \frac{P_T}{P_R}, \quad P_R = P_T \cdot 10^{-\frac{\gamma_0 L}{10}}. \quad (5.28)$$

This exponential dependence of loss vs. length is the same as for the transmission lines of Section 5.2.

- **Mode dispersion**, or the difference in group velocity between different modes, results in the broadening of a pulse which is launched into the fiber. This broadening of pulses results in interference between successive pulses which are transmitted, called *intersymbol interference* (Chapter 6). Since this pulse broadening increases with the length of the fiber, this dispersion will limit the distance between regenerative repeaters. One significant advantage of single mode fibers is that mode dispersion is absent since there is only one mode.
- **Chromatic or material dispersion** is caused by differences in the velocity of propagation at different wavelengths. For infrared and longer wavelengths, the shorter wavelengths arrive earlier than relatively longer wavelengths, but there is a crossover point at about  $1.3 \mu\text{m}$  beyond which relatively longer wavelengths arrive earlier. Since practical optical sources have a non-zero bandwidth, called the *linewidth*, and signal modulation increases the optical bandwidth further, material dispersion will also cause intersymbol interference and limit the distance between regenerative repeaters. Material dispersion is qualitatively similar to the dispersion that occurs in transmission lines (Section 5.2) due to frequency-dependent attenuation. The total dispersion is usually expressed in units of picoseconds pulse spreading per GHz source bandwidth per kilometer distance, with typical values in the range of zero to 0.15 in the  $1.3\text{--}1.6 \mu\text{m}$  minimum attenuation region [9,10]. It is very important that since the dispersion passes from positive to negative in the region of  $1.3 \mu\text{m}$  wavelength, the dispersion is very nearly zero at this wavelength. A typical curve of the magnitude of the chromatic dispersion vs. wavelength is shown in Figure 5-13, where the zero is evident. The chromatic dispersion can be made negligibly small over a relatively wide range of wavelengths. Furthermore, the frequency of this zero in chromatic dispersion can be shifted through waveguide design to correspond



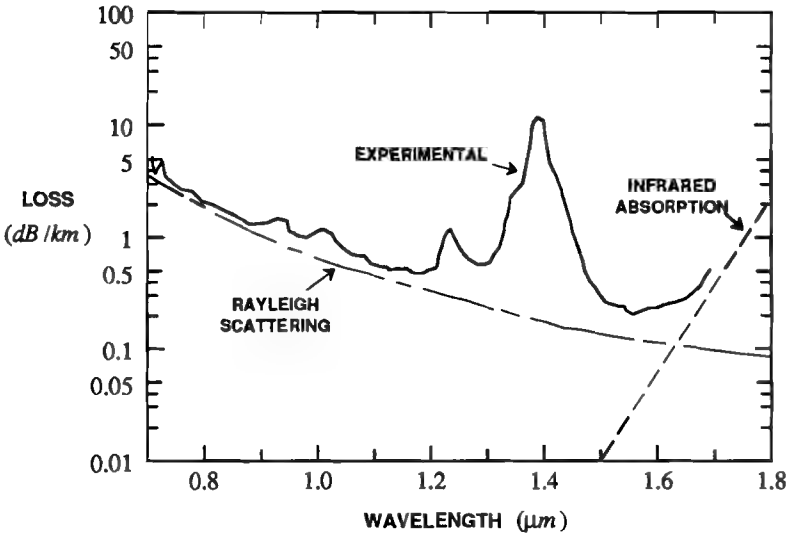
**Figure 5-13.** Typical chromatic dispersion in silica fiber [10]. Shown is the magnitude of the dispersion; the direction of the dispersion actually reverses at the zero-crossing.

to the wavelength of minimum attenuation.

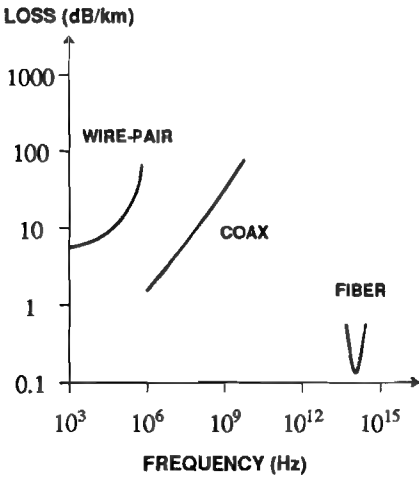
With these impairments in mind, we can discuss the practical and fundamental limits on information capacity for a fiber. The fundamental limit on attenuation is due to the intrinsic material scattering of the glass in the fiber — this is known as *Rayleigh scattering*, and is similar to the scattering in the atmosphere of the earth that results in our blue sky. The scattering loss decreases rapidly with wavelength (as the fourth power), and hence it is generally advantageous to choose a longer wavelength. The attenuation due to intrinsic absorption is negligible, but at certain wavelengths large attenuation due to certain impurities is observed. Particularly important are hydroxyl (OH) radicals in the glass, which absorb at 2.73  $\mu\text{meters}$  wavelength and harmonics. At long wavelengths there is infrared absorption associated fundamentally with the glass, which rises sharply starting at 1.6  $\mu\text{meters}$ .

A loss curve for a state-of-the-art fiber is shown in Figure 5-14. Note the loss curves for two intrinsic effects which would be present in an ideal material, Rayleigh scattering and infrared absorption, and additional absorption peaks at 0.95, 1.25, and 1.39  $\mu\text{m}$  due to OH impurities. The lowest losses are at approximately 1.3 and 1.5  $\mu\text{m}$ , and these are the wavelengths at which the highest performance systems operate. The loss is as low as about 0.2 dB/km, implying potentially a much larger repeater spacing for optical fiber digital communication systems as compared to wire-pairs and coax. A curve of attenuation vs. frequency in Figure 5-15 for wire cable media and for optical fiber illustrates that the latter has a much lower loss.

The loss per unit distance of the fiber is a much more important determinant of the distance between repeaters than is the bit rate at which we are transmitting. This is illustrated for a single-mode fiber in Figure 5-16, where there is an attenuation-limited region where the curve of repeater spacing vs. bit rate is relatively flat. As we increase the bit rate, however, we eventually approach a region where the repeater spacing is limited by the dispersion (mode dispersion in a multimode fiber and chromatic dispersion in a single mode fiber). The magnitude of the latter can be quantified simply by considering the Fourier transform of a transmitted pulse, and in particular its bandwidth  $W$ . The spreading of the pulse will be proportional to the

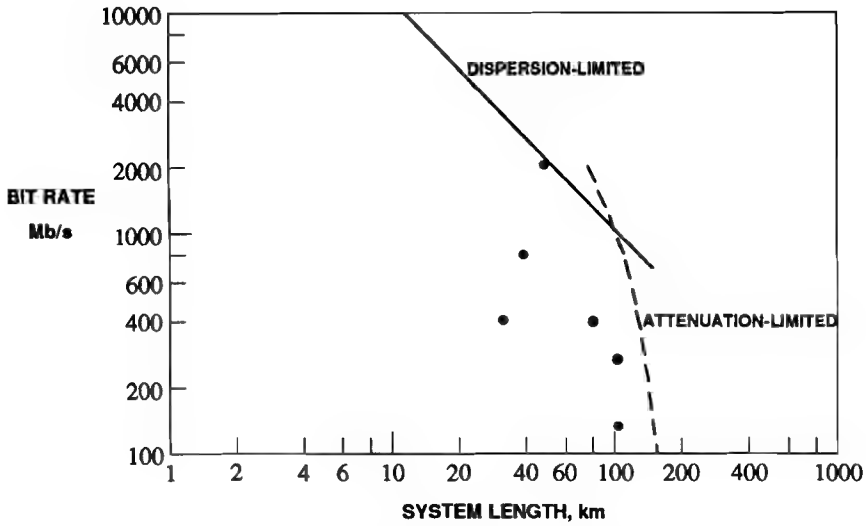


**Figure 5-14.** Observed loss spectrum of an ultra-low-loss germanosilicate single mode fiber together with the loss due to intrinsic material effects [11].



**Figure 5-15.** Attenuation vs. frequency for wire cable and fiber guiding media [10]. The band of frequencies over which the fiber loss is less than 1 dB/km is more than 10<sup>14</sup>Hz.





**Figure 5-16.** Tradeoff between distance and bit rate for a single mode fiber with a particular set of assumptions [8]. The dots represent performance of actual field trial systems.

repeater spacing  $L$  and the bandwidth  $W$ , with a constant of proportionality  $D$ . Thus, if we require that this dispersion be less than half a pulse-time at a pulse rate of  $R$  pulses per second,

$$D \cdot L \cdot W < \frac{1}{2R} . \quad (5.29)$$

The bandwidth  $W$  of the source depends on the *linewidth*, or intrinsic bandwidth in the absence of modulation, and also on the *modulation*. Since a non-zero linewidth will increase the bandwidth and hence the chromatic dispersion, we can understand fundamental limits by assuming zero linewidth. We will show in Chapter 6 that the bandwidth due to modulation is approximately equal to the pulse rate, or  $W \approx R$ , and hence (5.29) becomes

$$R^2 L < \frac{1}{2D} . \quad (5.30)$$

This equation implies that in the region where dispersion is limiting, the repeater spacing  $L$  must decrease rather rapidly as the bit rate is increased, as shown in Figure 5-16. More quantitative estimates of the limits shown in Figure 5-16 are derived in Problem 5-6 and Problem 5-11.

As a result of these considerations, first generation (about 1980) optical fiber transmission systems typically used multimode fiber at a wavelength of about 0.8  $\mu\text{meter}$ , and achieved bit rates up to about 150 Mb/s. The Rayleigh scattering is about 2 dB per km at this wavelength, and the distance between regenerative repeaters was in the 5 to 10 km range. Second generation systems (around 1985) moved to single mode fibers and wavelengths of about 1.3  $\mu\text{meters}$ , where Rayleigh scattering attenuation is about 0.2 dB/km (and practical attenuations are more on the order of 0.3

dB/km).

The finite length of manufactured fibers and system installation considerations dictate fiber connectors. These present difficult alignment problems, all the more difficult for single mode fibers because of the smaller core, but in practice connector losses of 0.1 to 0.2 dB can be obtained even for single mode fibers. Since it must be anticipated in any system installation that accidental breakage and subsequent splicing will be required at numerous points, in fact connector and splicing loss is the dominant loss in limiting repeater spacing.

Bending loss is due to the different propagation velocities required on the outer and inner radius of the bend. As the bending radius decreases, eventually the light on the outer radius must travel faster than the speed of light, which is of course impossible. What happens instead is that significant attenuation occurs due to a loss of confined power. Generally there is a tradeoff between bending loss and splicing loss in single mode fibers, since bending loss is minimized by confining most of the power to the core, but that makes splicing alignment more critical.

In Figure 5-16, the tradeoff between maximum distance and bit rate is quantified for a single mode fiber for a particular set of assumptions (the actual numerical values are dependent on these assumptions). At bit rates below about one Gb/s ( $10^9$  bits per second) the distance is limited by attenuation and receiver sensitivity. In this range the distance decreases as bit rate increases since the receiver sensitivity decreases (see Section 5.3.3). At higher bit rates, pulse broadening limits the distance before attenuation becomes important. The total fiber system capacity is best measured by a figure of merit equal to the product of the bit rate and the distance between repeaters (Problem 5-10), measured in Gb-km/sec. Current commercial systems achieve capacities on the order of 100 to 1000 Gb-km/sec.

### 5.3.2. Sources

While optical fiber transmission uses light energy to carry the information bits, at the present state of the art the signals are generated and manipulated electrically. This implies an electrical-to-optical conversion at the input to the fiber medium and an optical-to-electrical conversion at the output. There are two available light sources for fiber digital communication systems: the *light-emitting diode (LED)* and the *semiconductor injection laser*. The semiconductor laser is the more important for high-capacity systems, so we emphasize it here. In contrast to the LED, the laser output is *coherent*, meaning that it is nearly confined to a single frequency. In fact the laser output does have non-zero linewidth, but by careful design the linewidth can be made small relative to signal bandwidths using a structure called *distributed feedback (DFB)*. Thus, coherent modulation and demodulation schemes are feasible (Chapter 8), although commercial systems use intensity modulation. The laser output can be coupled into a single mode fiber with very high efficiency (about 3 dB power loss), and can generate powers in the 0 to 10 mW range [9] with one mW (0 dBm) typical [10]. The laser is necessary for single mode fibers, except for short distances, because it emits a narrower beam than the LED. The light output of the laser is very temperature dependent, and hence it is generally necessary to monitor the light output and control the driving current using a feedback circuit.

There is not much room for increasing the launched power into the fiber because of nonlinear effects which arise in the fiber [9], unless, of course, we can find ways to circumvent or exploit these nonlinear phenomena.

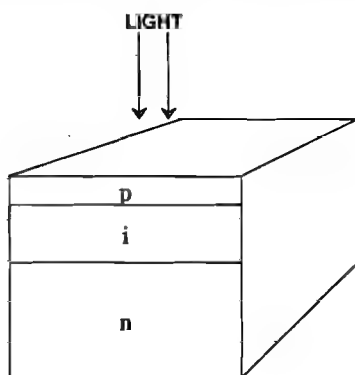
### 5.3.3. Photodetectors

The optical energy at the output of the fiber is converted to an electrical signal by a *photodetector*. There are two types of photodetectors available — the *PIN photodiode*, popular at about 100 Mb/s and below, and the *avalanche photodiode (APD)* (popular above 1 Gb/s) [12,10]. The cross-section of a PIN photodiode is shown in Figure 5-17. This diode has an intrinsic (non-doped) region (not typical of diodes) between the n- and p-doped silicon. Photons of the received optical signal are absorbed and create hole-electron pairs. If the diode is reverse biased, there is an electric field across the depletion region of the diode (which includes the intrinsic portion), and this electric field separates the holes from electrons and sweeps them to the contacts, creating a current proportional to the incident optical power. The purpose of the intrinsic region is to enlarge the depletion region, thereby increasing the fraction of incident photons converted into current (carriers created outside the depletion region, or beyond diffusion distance of the depletion region, recombine with high probability before any current is generated).

The fraction of incident photons converted into carriers that reach the electrodes is called the *quantum efficiency* of the detector, denoted by  $\eta$ . Given the quantum efficiency, we can easily predict the current generated as a function of the total incident optical power. The energy of one photon is  $h\nu$  where  $h$  is Planck's constant ( $6.6 \cdot 10^{-34}$  Joule-sec) and  $\nu$  is the optical frequency, related to the wavelength  $\lambda$  by

$$\nu\lambda = c \quad (5.31)$$

where  $c$  is the speed of light ( $3 \cdot 10^8$  m/sec). If the incident optical power is  $P$  watts,



**Figure 5-17.** A PIN photodiode cross-section. Electrode connection to the n- and p-regions creates a diode, which is reverse-biased.

then the number of photons per second is  $P/h\nu$ , and if a fraction  $\eta$  of these photons generate an electron with charge  $q$  ( $1.6 \cdot 10^{-19}$  Coulombs) then the total current is

$$i = \eta q \left( \frac{P}{h\nu} \right). \quad (5.32)$$

#### Example 5-15.

For a wavelength of  $1.5 \mu\text{m}$  and quantum efficiency of unity, what is the *responsivity* (defined as the ratio of output current to input power) for a PIN photodiode? It is

$$\frac{i}{P} = \frac{q}{h\nu} = \frac{q\lambda}{hc} = \frac{1.6 \cdot 10^{-19} \cdot 1.5 \cdot 10^{-6}}{6.6 \cdot 10^{-34} \cdot 3.0 \cdot 10^8} = 1.21 \text{ amps/watt}. \quad (5.33)$$

If the incident optical power is a nanowatt, the maximum current from a PIN photodiode is 1.21 nanoamperes.  $\square$

With PIN photodiodes (and more generally all photodetectors), there is a tradeoff between quantum efficiency and speed. Quantum efficiencies near unity are achievable with a PIN photodiode, but this requires a long absorption region. But a long intrinsic absorption region results in a correspondingly smaller electric field (with resulting slower carrier velocity) and a longer drift distance, and hence slower response to an optical input. Higher speed inevitably results in reduced sensitivity.

Since very small currents are difficult to process electronically without adding significant thermal noise, it is desirable to increase the output current of the diode before amplification. This is the purpose of the APD, which has internal gain, generating more than one electron-hole pair per incident photon. Like the PIN photodiode, the APD is also a reverse-biased diode, but the difference is that the reverse voltage is large enough that when carriers are freed by a photon and separated by the electric field they have enough energy to collide with the atoms in the semiconductor crystal lattice. The collisions ionize the lattice atoms, generating a second electron-hole pair. These secondary carriers in turn collide with the lattice, and additional carriers are generated. One price paid for this gain mechanism is an inherently lower bandwidth. A second price paid in the APD is the probabilistic nature of the number of secondary carriers generated. The larger the gain in the APD, the larger the statistical fluctuation in current for a given optical power. In addition, the bandwidth of the device decreases with increasing gain, since it takes some time for the avalanche process to build up.

Both PIN photodiodes and APD's exhibit a small current which flows in the absence of incident light due to thermal excitation of carriers. This current is called *dark current* for obvious reasons, and represents a background noise signal with respect to signal detection.

### 5.3.4. Model for Fiber Reception

Based on the previous background material and the mathematics of Poisson processes and shot noise (Section 3.4) we can develop a statistical model for the output of an optical fiber detector. This signal has quite different characteristics from that of other media of interest, since random quantum fluctuations in the signal are important. Since the signal itself has random fluctuations, we can consider it to have a

type of *multiplicative noise*.

In commercial systems, the *direct detection* mode of transmission is used, as pictured in Figure 5-18. In this mode, the intensity or power of the light is directly modulated by the electrical source (data signal), and a photodetector turns this power into another electrical signal. If the input current to the source is  $x(t)$ , then the output power of the source is proportional to  $x(t)$ .

Two bad things happen to this launched power as it propagates down the fiber. First, it is attenuated, reducing the signal power at the detector. Second, it suffers dispersion due to chromatic dispersion (and mode dispersion for a multimode fiber), which can be modeled as a linear filtering operation. Let  $g(t)$  be the impulse response of the equivalent dispersion filter, including the attenuation, so that the received power at the detector is

$$P(t) = x(t) * g(t). \quad (5.34)$$

In the final conversion to electrical current in the photodetector, the situation is a bit more complicated since quantum effects are important. The incident light consists of discrete photons which are converted to photoelectron-hole pairs in the detector. Hence, the current generated consists of discrete packets of charge generated at discrete points in time. Intuitively we might expect that the arrival times of the charge packets for a Poisson process (Section 3.4) since there is no reason to expect the interarrival times between photons to depend on one other. In fact, this is predicted by quantum theory. Let  $h(t)$  be the response of the photodetector circuit to a single photoelectron, and then an outcome for the detected current  $y(t)$  is a filtered Poisson process

$$Y(t) = \sum_m h(t - t_m) \quad (5.35)$$

where the  $t_m$  are Poisson arrival times. The Poisson arrivals are characterized by the rate of arrivals, which is naturally proportional to the incident power,

$$\lambda(t) = \frac{\eta}{h\nu} \cdot P(t) + \lambda_0 \quad (5.36)$$

where  $\eta$  is the quantum efficiency and  $\lambda_0$  is a dark current. Note from Campbell's theorem (Section 3.4.4) that the expected detected current is

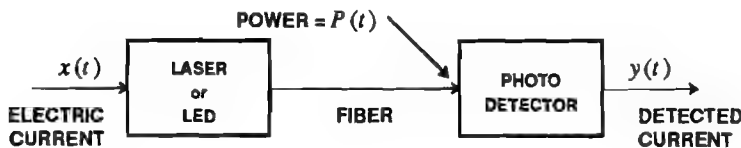


Figure 5-18. Elements of a direct detection optical fiber system.

$$E[Y(t)] = \lambda(t) * h(t) = \frac{\eta}{h\nu} \cdot x(t) * g(t) * h(t) + \lambda_0 H(0) . \quad (5.37)$$

The equivalent input-output relationship of the channel is therefore characterized, with respect to the mean-value of the detector output current, by the convolution of the two filters — the dispersion of the fiber and the response of the detector circuitry. Of course, there will be statistical fluctuations about this average that will be characterized in Chapter 8. This simple linear model for the channel that is quite accurate unless the launched optical power is high enough to excite nonlinear effects in the fiber and source-detector.

### Avalanche Photodetector

In the case of an APD, we have to modify this model by adding to the filtered Poisson process of (5.35) the random multiplier resulting from the avalanche process,

$$y(t) = \sum_m g_m h(t - t_m) , \quad (5.38)$$

the statistics of which has already been considered in (3.143). Define the mean and second moment of the avalanche gain,

$$\bar{G} = E[G_m] , \quad \overline{G^2} = E[G_m^2] . \quad (5.39)$$

Then from Section 3.4.6, we know that the effect of the avalanche gain on the second order statistics of (5.35) is to multiply the mean value of the received random process by  $\bar{G}$  and the variance by  $\overline{G^2}$ .

If the avalanche process were deterministic, that is precisely  $\bar{G}$  secondary electrons were generated for each primary photoelectron, then the second moment would be the square of the mean,

$$\overline{G^2} = \bar{G}^2 . \quad (5.40)$$

The effect of the randomness of the multiplication process is to make the second moment larger, by a factor  $F_G$  greater than unity,

$$\overline{G^2} = F_G \bar{G}^2 \quad (5.41)$$

where of course  $F_G = \overline{G^2}/\bar{G}^2$ . The factor  $F_G$  is called the *excess noise factor*. In fact, a detailed analysis of the physics of the APD [13] yields the result

$$F_G = k \cdot \bar{G} + (2 - \frac{1}{\bar{G}}) \cdot (1 - k) \quad (5.42)$$

where  $0 \leq k \leq 1$  is a parameter under the control of the device designer called the *carrier ionization ratio*. Note that as  $k \rightarrow 1$ ,  $F_G \rightarrow \bar{G}$ , or the excess noise factor is approximately equal to the avalanche gain. This says that the randomness gets larger as the avalanche gain gets larger. On the other hand, as  $k \rightarrow 0$ ,  $F_G \rightarrow 2$  for large  $\bar{G}$ , or the excess noise factor is approximately independent of the avalanche gain. Finally, when  $\bar{G} = 1$  (there is no avalanche gain),  $F_G = 1$  and there is no excess noise. This is the PIN photodiode detector.

### Fiber and Preamplifier Thermal Noise

Any physical system at non-zero temperature will experience noise due to the thermal motion of electrons, and optical fiber is no exception. This noise is often called *thermal noise* or *Johnson noise* in honor of J.B. Johnson, who studied this noise experimentally at Bell Laboratories in 1928. Theoretical study of this noise based on the theory of quantum mechanics was carried out by H. Nyquist at about the same time. Thermal noise is usually approximated as white Gaussian noise. The Gaussian property is a result of the central limit theorem and the fact that thermal noise is composed of the superposition of many independent actions. The white property cannot of course extend to infinite frequencies since otherwise the total power would be infinite, but rather this noise can be considered as white up to frequencies of 300 GHz or so. Nyquist's result was that thermal noise has an available noise power per Hz of

$$N(\nu) = \frac{h\nu}{e^{h\nu/kT_n} - 1} \quad (5.43)$$

where  $h$  is Planck's constant,  $\nu$  is the frequency,  $k$  is Boltzmann's constant ( $1.38 \cdot 10^{-23}$  Joules per degree Kelvin), and  $T_n$  is the temperature in degrees Kelvin. By *available noise power* we mean the power delivered into a load with a matched impedance. If we consider this as a two sided spectral density, we have to divide by two.

At frequencies up through the microwave, the exponent in (5.43) is very small, and if we approximate  $e^x$  by  $1 + x$  we get that the spectrum is approximately white,

$$N(\nu) \approx kT_n. \quad (5.44)$$

This corresponds to a two-sided spectral density of size

$$N_0 = \frac{kT_n}{2}. \quad (5.45)$$

However, at high frequencies, this spectrum approaches zero exponentially, yielding a finite total power.

There are two possible sources of thermal noise — at the input to the detector, and in the receiver preamplifier. At the input to the detector only thermal noise at optical frequencies is relevant (the detector will not respond to lower frequencies), and at these frequencies the thermal noise will be negligible.

#### Example 5-16.

At room temperature  $kT_n$  is  $4 \cdot 10^{-21}$  Joules. At 1 GHz, or microwave frequencies,  $h\nu$  is about  $10^{-24}$  Joules, and we are well in the regime where the spectrum is flat. However, at 1  $\mu\text{m}$  wavelength, or  $\nu = 3 \cdot 10^{14}$  Hz,  $h\nu$  is about  $2 \cdot 10^{-19}$  Joules, and  $\frac{h\nu}{kT_n}$  is about 50. Thus, the thermal noise is much smaller than  $kT_n$  at these frequencies. Generally thermal noise at optical frequencies is negligible in optical fiber systems at wavelengths shorter than about 2  $\mu\text{m}$  [14].  $\square$

Since the signal level is very low at the output of the detector in Figure 5-18, we must amplify the signal using a preamplifier as the first stage of a receiver. Thermal

noise introduced in the preamplifier is a significant source of noise, and in fact in many optical systems is the dominant noise source. Since the signal at this point is the baseband digital waveform, it occupies a bandwidth extending possibly up to microwave frequencies but not optical frequencies, hence the importance of thermal noise. We will see in Chapter 8 that this thermal noise is the primary reason for considering the use of an APD detector in preference to a PIN photodiode. A more detailed consideration of the design of the preamplifier circuitry is given in [14] and the problems in Chapter 8.

### 5.3.5. Advanced Techniques

Two exciting developments have been demonstrated in the laboratory: *soliton transmission* and *erbium-doped fiber amplifiers*. The soliton operates on the principle of the *optical Kerr effect*, a nonlinear effect in which the index of refraction of the fiber depends on the optical power. As previously mentioned, in chromatic dispersion, the index of refraction also depends on the wavelength. Solitons are optical pulses that have a precise shape and peak power chosen so that the Kerr effect produces a chirp (phase modulation) that is just appropriate to cancel the pulse broadening induced by group-velocity dispersion. The result is that all the wavelengths can be made to travel at the same speed, essentially eliminating material dispersion effects. In soliton transmission, material attenuation is the only effect that limits repeater spacing.

An optical amplifier can be constructed out of a fiber doped with the rare-earth element erbium, together with a semiconductor laser pumping source. If the pumping source wavelength is 0.98 or 1.48  $\mu\text{m}$ , then the erbium atoms are excited into a higher state, and reinforce 1.55  $\mu\text{m}$  incident light by stimulated emission. With about 10 mW of pumping power, gains of 30 to 40 dB at 1.55  $\mu\text{m}$  can be obtained. A receiver designed using this optical amplifier is shown in Figure 5-19. The optical amplifier has gain  $G$ , which actually depends on the input signal power because large signals deplete the excited erbium atoms and thereby reduce the gain. The amplifier also generates a spurious noise due to spontaneous emission, and the purpose of the optical bandpass filter is to filter out spontaneous noise outside the signal bandwidth (which depends on source linewidth as well as signal modulation). There is a premium on narrow linewidth sources, because that enables the optical filter bandwidth to be minimized.

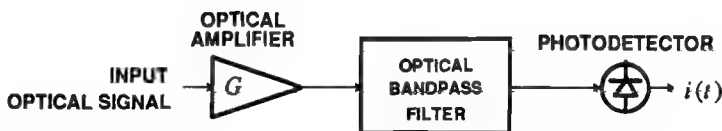


Figure 5-19. A direct-detection optical receiver using an optical amplifier.



The effect of the amplifier is similar to an avalanche detector, in that it increases the signal power (rendering electronic thermal noise insignificant) while adding additional spontaneous noise. The major distinction between the amplifier and avalanche detector, however, is that much of the spontaneous noise in the amplifier can be optically filtered out, whereas in the detector it cannot. It is also possible to place optical amplifiers at intermediate points in a fiber system, increasing the repeater spacing dramatically. The design of the receiver in Figure 5-19 will be considered further in Chapter 8.

## 5.4. MICROWAVE RADIO

The term "radio" is used to refer to all electromagnetic transmission through free space at microwave frequencies and below. There are many applications of digital transmission which use this medium, primarily at microwave frequencies, a representative set of which include *point-to-point terrestrial digital radio*, *digital mobile radio*, *digital satellite communication*, and *deep-space digital communication*.

Terrestrial digital radio systems use microwave horn antennas placed on towers to extend the horizon and increase the antenna spacing. This medium has been used in the past principally for analog transmission (using FM and more recently SSB modulation), but in recent years has gradually been converted to digital transmission due to increased demand for data services.

### Example 5-17.

In North America there are frequency allocations for telephony digital radios centered at frequencies of 2, 4, 6, 8, and 11 GHz [15]. In the United States there were over 10,000 digital radio links in 1986, including a cross-country network at 4 GHz. □

A related application is digital mobile radio.

### Example 5-18.

The frequency band from 806 to 947 MHz is allocated in the United States to land mobile radio services [16]. This band is used for *cellular mobile radio* [17], in which a geographical area is divided into a lattice of cells, each with its own fixed omni-directional base antenna for transmission and reception. As a vehicle passes through the cells, the associated base antenna is automatically switched to the closest one. An advantage of this concept is that additional mobile telephones can be accommodated by decreasing the size of the cells and adding additional base antennas. □

Satellites are used for long-distance communication between two terrestrial antennas, where the satellite usually acts as a non-regenerative repeater. That is, the satellite simply receives a signal from a terrestrial transmitting antenna, amplifies it, and transmits it back toward another terrestrial receiving antenna. Satellite channels offer an excellent alternative to fiber and cable media for transmission over long distances, and particularly over sparse routes where the total communication traffic is small. Satellites also have the powerful characteristic of providing a natural multiple access medium, which is invaluable for random-access communication among a

number of users. A limitation on satellites is limited power available for transmission, since the power is derived from solar energy or expendable resources. In addition, the configuration of the launch vehicles usually limit the size of the transmitting and receiving antennas (which are usually one and the same). Most communication satellites are put into synchronous orbits, so that they appear to be stationary over a point on the earth. This greatly simplifies the problems of antenna pointing and satellite availability.

In deep-space communication, the object is to transmit data to and receive data from a platform that is at a great distance from earth. This application includes the features of both satellite and mobile communication, in that the vehicle is usually in motion. As in the satellite case, the size of the antenna and the power resources at the space vehicle are limited.

With the exception of problems of multipath propagation in terrestrial links, the microwave transmission channel is relatively simple. There is an attenuation introduced in the medium due to the spreading of the energy, where this attenuation is frequency-independent, and thermal noise introduced at the antenna and in the amplifiers in the receiver. These aspects of the channel are covered in the following subsections followed by a discussion of multipath distortion.

### 5.4.1. Microwave Antennas and Transmission

Microwave propagation through free-space is very simple, as there is an attenuation due to the spreading of radiation. The attenuation varies so slowly with frequency that it can be considered virtually fixed within the signal bandwidth. Consider first an *isotropic antenna*; namely, one that radiates power equally in all directions. Assume the total radiated power is  $P_T$  watts, and assume that at distance  $d$  meters from this transmit antenna there is a receive antenna with area  $A_R$  meters<sup>2</sup>. Then the maximum power that the receive antenna could capture is the transmit power times the ratio of  $A_R$  to the area of a sphere with radius  $d$ , which is  $4\pi d^2$ . There are two factors which modify this received power. First, the transmit antenna can be designed to focus or concentrate its radiated energy in the direction of the receiving antenna. This adds a factor  $G_T$  called the *transmit antenna gain* to the received power. The second factor is the *antenna efficiency*  $\eta_R$  of the receive antenna, a number less than (but hopefully close to) unity; the receive antenna does not actually capture all the electromagnetic radiation incident on it. Thus, the received power is

$$P_R = P_T \frac{A_R}{4\pi d^2} G_T \eta_R. \quad (5.46)$$

At microwave frequencies, aperture antennas (such as horn or parabolic) are typically used, and for these antennas the achievable antenna gain is

$$G = \frac{4\pi A}{\lambda^2} \eta, \quad (5.47)$$

where  $A$  is the area of the antenna,  $\lambda$  is the wavelength of transmission, and  $\eta$  is an efficiency factor. Expression (5.47) applies to either a receiving or transmitting antenna, where the appropriate area and efficiency are substituted. This expression is

intuitively pleasing, since it says that the antenna gain is proportional to the square of the ratio of antenna dimension to wavelength. Thus, the transmit antenna size in relation to the wavelength is all that counts in the directivity or gain of the antenna. This antenna gain increases with frequency for a given antenna area, and thus higher frequencies have the advantage that a smaller antenna is required for a given antenna gain. The efficiency of an antenna is typically in the range of 50 to 75 percent for a parabolical reflector antenna and as high as 90 percent for a horn antenna [18].

An alternative form of the received power equation can be derived by substituting for the area of the receive antenna in (5.46) in terms of its gain in (5.47),

$$\frac{P_R}{P_T} = G_T G_R \left[ \frac{\lambda}{4\pi d} \right]^2; \quad (5.48)$$

this is known as the *Friis transmission equation*. The term in brackets is called the *path loss*, while the terms  $G_T$  and  $G_R$  summarize the effects of the two antennas. While this loss is a function of wavelength, the actual power over the signal bandwidth does not generally vary appreciably where the bandwidth is very small in relation to the center frequency of the modulated signal. The Friis equation does not take into account other possible sources of loss such as rain attenuation and antenna mispointing.

The application of these relations to a particular configuration can determine the received power and the factors contributing to the loss of signal power. This process is known as generating the *link power budget*, as illustrated by the following examples.

#### Example 5-19.

Determine the received power for the link from a synchronous satellite to a terrestrial antenna for the following parameters: Height 40,000 km, satellite transmitted power 2 watts, transmit antenna gain 17 dB, receiving antenna area 10 meters<sup>2</sup> with perfect efficiency, and frequency 11 GHz. The wavelength can be obtained from (5.31),

$$\lambda = \frac{c}{v} = \frac{3 \cdot 10^8}{11 \cdot 10^9} = 27.3 \text{ mm}. \quad (5.49)$$

The receive antenna gain is

$$10\log_{10} G_R = 10\log_{10} \frac{4\pi \cdot 10}{(27.3 \cdot 10^{-3})^2} = 52.2. \quad (5.50)$$

Next, the path loss is

$$10\log_{10} \left[ \frac{\lambda}{4\pi d} \right]^2 = 20\log_{10} \left[ \frac{27.3 \cdot 10^{-3}}{4\pi \cdot 4 \cdot 10^7} \right] = -205.3 \text{ dB}. \quad (5.51)$$

Finally, we are in a position to calculate the received power, which we express in dBW (decibels relative to one watt) and recognizing that the transmit power is 3 dBW,

$$10\log_{10} P_R = 3 \text{ dBW} + 17 + 52.2 - 205.3 = -133 \text{ dBW}. \quad (5.52)$$

□

**Example 5-20.**

The Mariner-10 deep-space mission to Mercury in 1974 used a transmitter power of 16.8 watts and frequency 2.3 GHz. The transmit antenna diameter on the spacecraft was 1.35 meters with efficiency 0.54, which results in an antenna gain of 27.6 dB. The terrestrial receive antenna diameter was 64 meters with efficiency 0.575, for an antenna gain of 61.4 dB. The distance from the spacecraft to ground was  $1.6 \cdot 10^{11}$  meters, for a path loss of 263.8 dB. Finally, the received power was

$$10 \log_{10} P_R = 10 \log_{10} 16.8 + 27.6 + 61.4 - 263.8 = -162.6 \text{ dBW}. \quad (5.53)$$

□

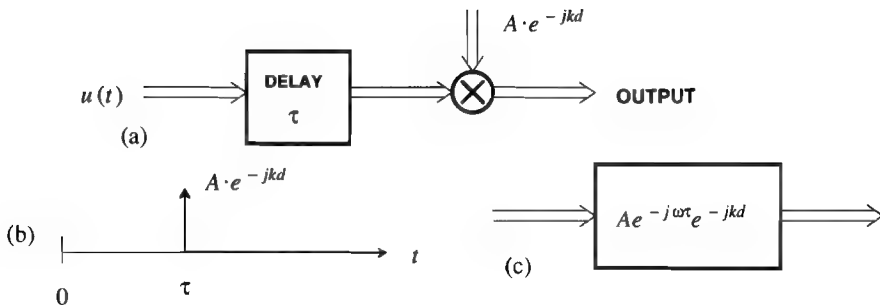
The two dominant effects of microwave propagation are attenuation and delay. It is of interest to determine the effect on a passband signal, represented by the complex-baseband signal of Section 2.4. Assume the attenuation is  $A$ , the distance of propagation is  $d$ , and the speed of propagation is  $c$ . The delay the signal experiences is  $\tau = d/c$ , and given a passband signal of the form of Figure 2-6, the output of the channel is

$$\sqrt{2}A \cdot \text{Re}\{ u(t - \tau) e^{j\omega_c(t - \tau)} \} = \sqrt{2} \cdot \text{Re}\{ A u(t - \tau) e^{-jkd} e^{j\omega_c t} \} \quad (5.54)$$

where

$$k = \frac{\omega_c \tau}{d} = \frac{\omega_c}{c} = \frac{2\pi}{\lambda}, \quad (5.55)$$

is called the *propagation constant*. The equivalent complex-baseband channel, shown in Figure 5-20, characterizes the effect of propagation on the equivalent complex-baseband signal. Not surprisingly, the baseband signal is delayed by  $\tau$ , the same as the passband signal. In addition, there is a phase shift by  $kd = 2\pi d/\lambda$  radians, or  $2\pi$  radians for each wavelength of propagation distance. The equivalent complex-valued impulse response of the propagation is an impulse with delay  $\tau$  and area  $A \cdot e^{-jkd}$ , and the equivalent transfer function is  $A e^{-j\omega\tau} e^{-jkd}$ . The only frequency dependence is



**Figure 5-20.** The equivalent complex baseband channel for a freespace propagation with attenuation  $A$  and distance  $d$ . a) Equivalent system, b) the equivalent impulse response, and c) the equivalent baseband transfer function.

linear in frequency, due to the delay. For mobile receivers, the effect of small changes in  $d$  on the baseband channel response is particularly significant. The effect is dramatically more pronounced for the phase shift than for the delay.

#### Example 5-21.

For a carrier frequency of 1 GHz (typical for mobile radio), the propagation constant is  $k = \omega_c/c = 21$  radians/meter. Thus, a  $\pi/2 = 1.57$  radian phase shift, which will be very significant to demodulators, occurs with every 7.4 centimeters change in propagation distance. In contrast, the propagation delay changes by 3.3 nanoseconds for each meter change in propagation distance. In relation to typical baseband bandwidths, this is totally insignificant. For example, at 1 MHz, the change in phase shift due to this delay change is only  $\omega\tau = 2\pi \cdot 0.0033$ , or roughly one degree.  $\square$

### 5.4.2. Noise in Microwave Amplifiers

On a radio link noise enters the receiver both through the antenna and as internal noise sources in the receiver. We saw in (5.45) that both sources of noise are Gaussian and can be considered as white up through the microwave frequencies. White noise is completely specified by the spectral density  $N_0$ , given by (5.45). However, in radio transmission it is common to express this spectral density in terms of an equivalent parameter, the *noise temperature* expressed in degrees Kelvin. This custom derives from the functional form of (5.45), where  $N_0$  is strictly a function of the temperature. The custom of specifying noise temperature derives from the fact that  $T_n$  is reasonable in size, on the order of tens or hundreds of degrees, whereas  $N_0$  is a very small number. Note however that the total thermal noise at some point in a system may be the superposition of many thermal noise sources at different temperatures. Hence, the noise temperature is merely a convenient specification of the noise power, and is not necessarily equal to the physical temperature of any part of the system! For example, if we amplify the noise we increase the noise temperature without affecting the physical temperature of the source that generated that noise.

There are two sources of noise — the noise incident on the antenna and the noise introduced internally in the receiver. The noise incident on the antenna depends on the effective noise temperature in the direction the antenna is pointed. For example, the sun has a much higher effective temperature than the atmosphere. The noise introduced internal to the receiver depends on the design and sophistication of the receiver. It is customary to refer all noise sources to the input to the receiver (the antenna), and define an equivalent noise temperature at that point. Since the stages of a receiver typically have large gains, the noise introduced internal to the receiver usually has a much smaller noise temperature when referred to the receiver input. These receiver noise temperatures range from about four degrees Kelvin for supercooled maser amplifiers to the range of 70 to 200 degrees Kelvin for receivers without physical cooling.

#### Example 5-22.

Continuing Example 5-20 for the Mariner-10 mission the effective noise temperature of the antenna plus receiver was 13.5 degrees Kelvin. A bit rate of 117.6 kb/s was used. What is the signal-to-noise ratio in the receiver assuming the bandwidth of the system is half the bit

rate, 58.8 kHz? (We will see in Chapter 6 that this is the minimum possible bandwidth for binary transmission.) The total noise power within the receiver bandwidth would be

$$P_n = kT_n B = 1.38 \cdot 10^{-23} \cdot 13.5 \cdot 58.8 \cdot 10^3 = -169.6 \text{ dBW} \quad (5.56)$$

The signal-to-noise ratio is therefore

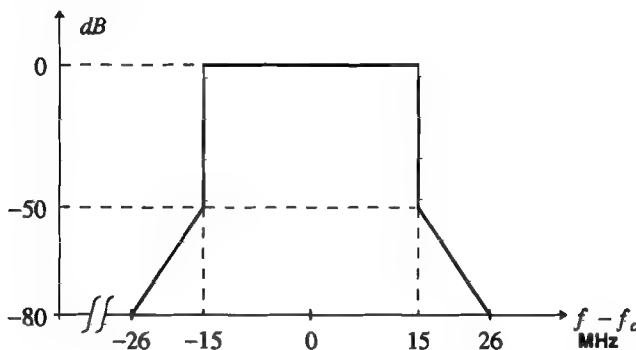
$$SNR = -162.6 + 169.6 = 7.0 \text{ dB}. \quad (5.57)$$

In practice the noise bandwidth will be larger than this, and the SNR will be lower, perhaps by a couple of dB. This SNR will support data transmission, albeit at a rather poor error rate. Coding techniques (Chapters 13 and 14) can compensate for the poor SNR.  $\square$

The SNR as calculated in this example is a useful quantity since it will not be changed by the large gain introduced in the receiver (both signal and noise will be affected the same way).

### 5.4.3. Emission Masks

A radio channel does not in itself provide any significant restriction on the bandwidth that we can use for digital communication. Moreover, the free-space channel introduces only a slowly varying dependence of attenuation on frequency (due to antenna gain). Thus, there is nothing inherent about the channel to introduce significant motivation to be spectrally efficient. Enter the regulatory agencies! Since the radio spectrum is a scarce commodity, it is carefully allocated to individual users. Unlike optical-fiber, where different users can install their own fiber, we must all share a single radio environment. To prevent significant interference between users, *spectral emission masks* are specified by regulation. An example of such a mask is shown in Figure 5-21. In this case, the regulatory agency has assigned a nominal 30 MHz bandwidth centered at  $f_c$  to a particular user, but for practical reasons has allowed that user to transmit a small amount of power (down more than 50 dB)



**Figure 5-21.** A spectral emission mask referenced to a nominal 30 MHz channel bandwidth. The vertical axis is transmitted power spectrum referenced to the power of an unmodulated carrier. The user signal must stay under the mask. (This mask applies to the United States.)

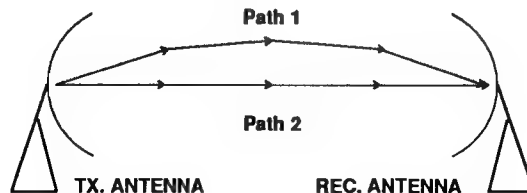
outside that band.

This mask is usually adhered to by placing a very sharp cutoff filter in the radio transmitter. Since this filter is imposed by external constraints, it is natural to think of this filter as being part of the channel (this logic is oversimplified of course since the filter requirements depend on the spectrum of the signal feeding the filter). From this perspective, the microwave radio channel has a very sharp cutoff at the band edges, in contrast to the media we have seen earlier which have at most a gradual increase of attenuation with frequency. This characteristic is shared by the voiceband data channel in the next section, and for this reason similar modulation techniques are often employed on the two media.

#### 5.4.4. Multipath Fading

We have seen how the link budget can be determined for a radio link. The calculation we made assumed for the most part idealized circumstances, whereas in practice additional *system margin* must be included in the link budget to account for foreseen or unforeseen circumstances. For example, at higher frequencies there will be an additional *rain attenuation* during rainstorms at the earth receiving antenna. In terrestrial microwave systems, there is an additional important source of attenuation that must be accounted for — *multipath fading* [19]. Both rain attenuation and multipath fading result in an attenuation on the signal path that varies with frequency. A significant difference is that unlike rain attenuation, multipath fading can result in a large frequency-dependent attenuation within the narrow signal bandwidth. This phenomenon is known as *selective fading*.

The mechanism for multipath fading, shown in Figure 5-22, is very similar to mode distortion in multimode optical fibers and to the distortion introduced by bridged taps in wire-pairs, except that it is time varying. The atmosphere is inhomogeneous to electromagnetic radiation due to spatial variations in temperature, pressure, humidity, and turbulence. This inhomogeneity results in variations in the index of refraction, resulting in possibly two or more ray paths for electromagnetic waves to travel from transmitter to receiver. Another source of multipath is the reflection of radio waves off of obstacles, such as buildings. The effective path lengths may be different for the different rays, and in general will interfere with one another since the receiver will perceive only the sum of the signals.



**Figure 5-22.** Illustration of two ray paths between a transmit and receive radio antenna. Fading attenuation results when the two paths have different propagation delays.

We can determine the effect of multipath fading on a passband signal using the equivalent complex-baseband response for a single path and applying superposition. If we assume two paths have attenuations  $A_1$  and  $A_2$  and propagation distances  $d_1$  and  $d_2$ , corresponding to propagation delays  $\tau_1 = d_1/c$  and  $\tau_2 = d_2/c$ , we can define two parameters  $\Delta d = d_1 - d_2$  and  $\Delta\tau = \tau_1 - \tau_2$ . Then by superposition the equivalent complex-baseband channel transfer function is

$$\begin{aligned} & A_1 e^{-j\omega\tau_1} e^{-jk d_1} + A_2 e^{-j\omega\tau_2} e^{-jk d_2} \\ &= A_1 e^{-j\omega\tau_1} e^{-jk d_1} \left( 1 + \frac{A_2}{A_1} e^{j\omega\Delta\tau} e^{jk\Delta d} \right). \end{aligned} \quad (5.58)$$

The first terms have a constant and linear phase shift due to the delay  $\tau_1$ , identical to the first path. The term in parentheses is important, because it can display a complicated dependence on frequency due to constructive and destructive interference of the two signals at the receiver.

The critically important parameter is  $\Delta\tau$ , which is called the *delay spread*. Two distinct cases can be distinguished. The first occurs when, for baseband frequencies of interest,  $|\omega\Delta\tau| \ll \pi$ , so that the frequency-dependence of the second term is insignificant. This is called the *narrowband model*. For this case, the two path propagation is similar to a single path, in that it results in a delay (linear phase shift with frequency) plus a constant phase shift. The contrary case is called the *broadband model*, and results in a more complicated frequency dependence due to constructive and destructive interference.

#### Example 5-23.

Assume that we define the transition between the narrowband and broadband model as a delay spread such that  $|\omega\Delta\tau| = 0.01\pi$  (1.8 degrees) at the highest baseband frequency of interest. Equivalently, we expect that  $f = 1/200\Delta\tau$  for the highest frequency. Then if the delay spread is 1 nanosecond, baseband channels with a bandwidth less than 5 MHz are considered narrowband, and bandwidths greater than 5 MHz (especially those significantly greater) are considered broadband. If the delay spread increases to 100 nanoseconds, then the narrowband channel has bandwidth less than 50 kHz according to this criterion. Note that all that counts is the delay spread, and not the absolute delay nor the carrier frequency. Also note that the actual passband signal has a bandwidth double the equivalent complex baseband signal.  $\square$

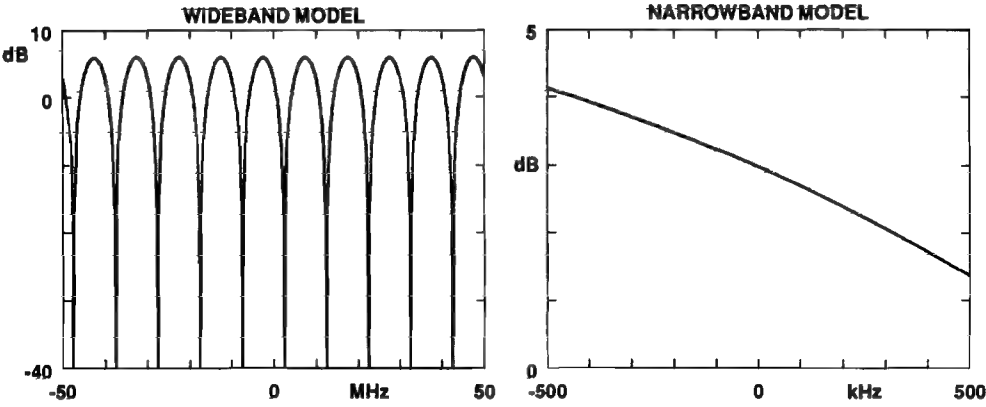
#### Example 5-24.

For the two-path case, the magnitude-squared of the frequency response for the frequency-dependent term of interest is

$$|1 + \rho e^{j\omega\Delta\tau}|^2 = 1 + |\rho|^2 + 2\operatorname{Re}\{\rho e^{j\omega\Delta\tau}\} \quad (5.59)$$

for some complex constant  $\rho$ . We will choose a delay spread of 10 nanoseconds (a typical worst-case number in an urban environment) and a fairly large  $|\rho| = 0.99$ . This is plotted in dB in Figure 5-23 over a  $\pm 50$  MHz frequency range, a broadband model, and a narrower frequency range, a narrowband model. Note the large notches due to destructive interference at some frequencies, accentuated by the fact that the two paths are nearly the same amplitude. Also note the close to 6 dB gain at some frequencies due to constructive interference. The narrowband model is plotted over a  $\pm 500$  kHz frequency range, which by



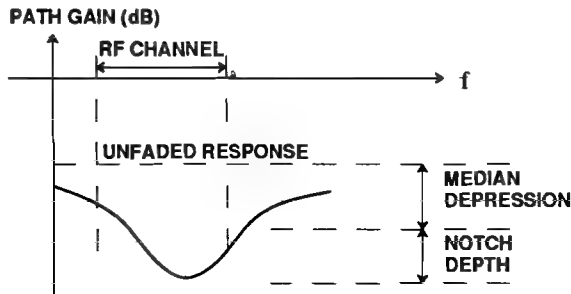


**Figure 5-23.** Complex baseband channel amplitude response over a wide frequency range and a narrow frequency range for a two-path model with  $p = 0.99j$ .

the criterion of Example 5-23 is a narrowband model. Note that the channel response varies only a couple of dB over this range. □

The two-path model, which is usually adequate for fixed terrestrial microwave systems, suggests that fading may result in either a monotonic gain change (or slope) across the channel or as a dip (or notch) in the channel response within the bandwidth. A typical faded channel response is shown in Figure 5-24, and the typical parameters that characterize the fade are identified [15].

In Section 5.4.1, we showed that the power loss in freespace radio propagation obeys a square-law relationship; that is, the receive power decreases as  $d^{-2}$ , or the path loss in dB increases as  $20 \cdot \log_{10} d$ . For terrestrial microwave transmission, the path loss increases more rapidly than in freespace, typically more like  $d^{-4}$  or



**Figure 5-24.** A typical frequency-selective notch due to fading with some terminology. Note that the impact on the channel depends strongly on the location of the notch relative to the channel bandwidth.

$40 \cdot \log_{10} d$  in dB. This can be explained using the simple model of Figure 5-25. Even for highly directional antennas, for a large  $d$  there will be a substantial reflection off the ground interfering at the receive antenna. Typically the ground is close to a short circuit for oblique angles of incidence at microwave frequencies, implying a reflection coefficient near  $-1$  (so that the net incident and reflected electric fields sum to zero).

### Exercise 5-3.

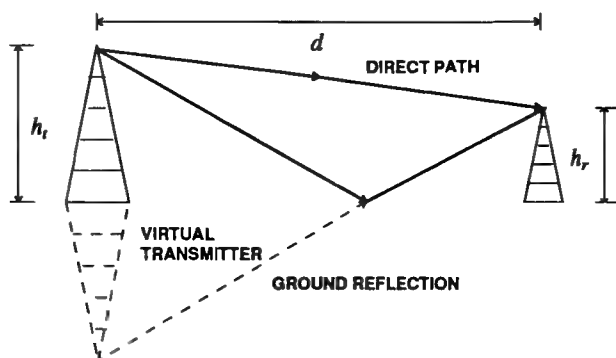
For the geometry of Figure 5-25, consider only the reflection resulting if the ground acts like a perfect mirror, and that both the direct and indirect paths suffer a freespace loss. Assuming the distance between antennas is much higher than the antenna heights, show that the resulting net power loss is approximately

$$\frac{P_R}{P_T} = \left( \frac{P_R}{P_T} \right)_{\text{freespace}} \left[ \frac{4\pi h_t h_r}{\lambda d} \right]^2. \quad (5.60)$$

Hence, the effect of the reflection is a destructive interference that increases the path loss by another factor of  $d^{-2}$  over and above the freespace loss.  $\square$

Note that, not unexpectedly, it is advantageous to have high antennas (the loss decreases as the square of the antenna heights).

Even when the transmitter and receiver are at fixed locations relative to one another, fading is a time-varying phenomenon for large distances (30 km or greater) due to atmospheric phenomena. Of considerable importance to designers of radio systems is not only the depth but also the duration of fades. Fortunately, it has been observed that the deeper the fade, the less frequently it occurs and the shorter its duration when it does occur. Also, the severity of fades increases as the distance between antennas increases or as the carrier frequency increases. Fading can also be mitigated



**Figure 5-25.** The attenuation of a terrestrial microwave system is increased by the ground reflection. There will be ground reflections from all points between the two antennas, but the single reflection resulting if the ground acts like a perfect mirror is shown. The transmit antenna height is  $h_t$ , the receive antenna height is  $h_r$ , and the distance between antennas is  $d$ .

by using *diversity techniques*, in which two or more independent channels are somehow combined [20]. The philosophy here is that only one of these channels at a time is likely to be affected by fading.

### 5.4.5. Mobile Radio

One of the most appealing uses for radio transmission is for communication with people or vehicles on the move. For this type of communication there is really no alternative to radio transmission, except for infrared, which does not work well outdoors. Mobile radio exhibits some characteristics that are different from point-to-point transmission. First, antennas must generally be omni-directional, and thus they exhibit much less antenna gain. Second, there can be obstacles to direct propagation, causing a *shadowing effect* that results in large variations in received signal power with location. Third, the most common application is in urban areas, where there are many opportunities for multiple reflections, and the two-path model is usually not accurate. Fourth, the user is often moving, resulting in extreme time-variations in transmission conditions over even short distances, as well as Doppler shift in the carrier frequency.

The two-path model is easily extended to an  $M$ -path model, again using superposition. In this case, the complex-baseband output of the channel is

$$\sum_{i=1}^M A_i u(t - \tau_i) e^{-jkd_i} \quad (5.61)$$

where the  $A_i$  are real-valued attenuation coefficients,  $d_i$  is the length of the  $i$ -th path, and  $\tau_i$  is the propagation delay of the  $i$ -th path. There may be a dominant path whose attenuation coefficient obeys the fourth-power law with distance, but the other coefficients depend on the reflection coefficients of indirect paths and hence bear a complicated relationship to position. Furthermore, due to shadow effects, there may even be no dominant path. For example if the mobile receiver is located behind a building; the radio waves will suffer a diffraction loss. This shadowing loss typically varies markedly over a distance of tens to hundreds of meters. If we average the received power over an area on the order of  $1 \text{ km}^2$ , we will see the fourth-power loss with distance, but if we average over an area on the order of  $1 \text{ meter}^2$  we will see an additional fluctuation with position due to shadowing. Shadowing is often assumed to result in a log-normal distribution in local-average received power; that is, the power expressed in dB has a Gaussian distribution. The standard deviation of the power expressed in dB is roughly 4 dB for typical urban areas.

When we examine local received power, not averaged over an area, we begin to see wild fluctuations due to multipath fading. For a moving vehicle, fades of 40 dB and more below the local-average level are frequent, with successive minima occurring every half wavelength or so (a fraction of a meter at microwave frequencies). Thus, the motion of the vehicle introduces a whole new dimension to the fading experienced on a point-to-point system, where the fluctuations are much slower. This rapid fluctuation is known as *Rayleigh fading* because the distribution of the envelope of the received carrier often obeys a Rayleigh distribution [21].

To understand Rayleigh fading, we must examine the effect of vehicle motion, which results in a time variation in received carrier phase. As before, this can be understood by considering a single path, and then applying superposition to multiple paths. The geometry of a single path is shown in Figure 5-26, including a reflection between the transmitter and receiver. As shown, a virtual transmitter can be defined behind the reflector with a linear propagation to the receiver. Let  $\mathbf{d}$  be a vector from virtual transmitter to receiver at time  $t = 0$ , let  $\mathbf{v}$  be the velocity vector for the vehicle at time  $t = 0$ , and let  $\theta$  be the angle between  $\mathbf{d}$  and  $\mathbf{v}$ , or the angle of incidence of the propagation path relative to the vehicle velocity. Let the scalar initial distance and velocity be  $d = \|\mathbf{d}\|$  and  $v = \|\mathbf{v}\|$ . The vector from transmitter to receiver is  $\mathbf{d} + \mathbf{v} \cdot t$ , and the propagation distance as a function of time is

$$\|\mathbf{d} + \mathbf{v} \cdot t\| = \left[ d^2 + v^2 t^2 + 2\langle \mathbf{d}, \mathbf{v} \rangle t \right]^{1/2} \quad (5.62)$$

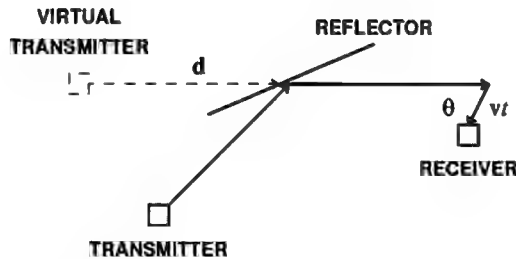
where the inner product is  $\langle \mathbf{d}, \mathbf{v} \rangle = dv \cdot \cos\theta$ . This distance is not changing linearly with time, but can be approximated by a linear function of time.

**Exercise 5-4.**

Show that if  $t \ll d/v$ , then (5.62) can be approximated accurately by  $d + vt \cdot \cos\theta$ . For example, if  $d = 1$  km and  $v = 30$  m/sec (approximately 100 km/hr) then the approximation holds for  $t \ll 66$  sec.  $\square$

The time scale over which the linear approximation to distance is valid is quite large relative to the significant carrier phase fluctuations, and hence it is safe to assume that the distance to the receiver is changing as  $v \cdot \cos\theta \cdot t$ . This change in distance has slope  $+v$  when the receiver is moving directly away from the transmitter,  $-v$  when it is moving directly toward the transmitter, and zero when the receiver is moving orthogonally to the transmitter.

With this basic geometric result in hand, the received complex-baseband signal is



**Figure 5-26.** Trajectory of motion for a vehicle moving at constant velocity, relative to a propagation path including a reflection.

$$A \cdot \text{Re} \left\{ u \left( t - \frac{d}{c} - \frac{v}{c} \cos \theta t \right) e^{-jkd} e^{-jkv \cos \theta t} e^{j\omega_c t} \right\} . \quad (5.63)$$

We see here several propagation effects. First, the baseband signal  $u(t)$  is delayed by a time-varying amount, due to the changing propagation distance. This effect is generally insignificant at the baseband frequencies of interest. Second, there is a static phase shift  $e^{-jkd}$  due to the propagation distance at  $t = 0$ . Third, and most interesting, is a phase shift that is linear with time. In effect, this is a frequency offset, known as the *Doppler shift*. The carrier frequency is shifted from  $\omega_c$  to  $\omega_c - \omega_d$ , where the Doppler frequency is

$$\omega_d = kv \cos \theta = \frac{2\pi v}{\lambda} \cos \theta . \quad (5.64)$$

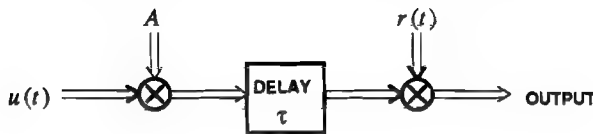
When the receiver is moving away from the transmitter, the Doppler shift is negative; it is positive when the receiver is moving toward the transmitter.

#### Example 5-25.

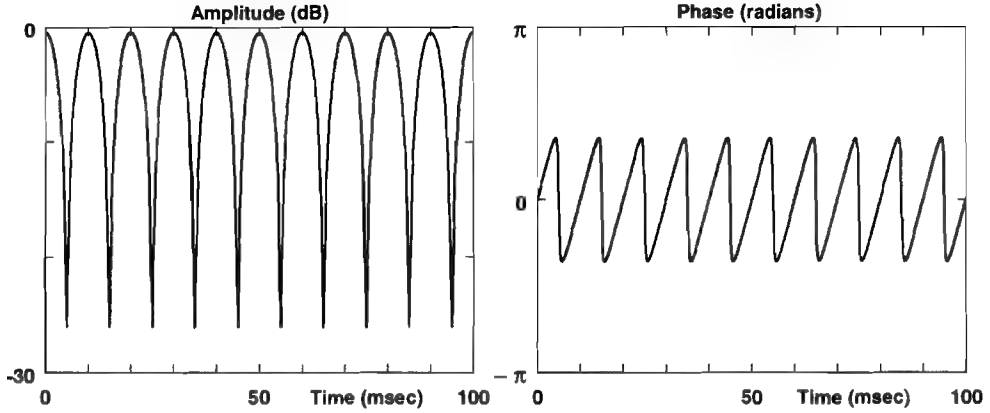
If the vehicle velocity is  $v = 30$  m/sec (100 km/hr), and the carrier frequency is 1 GHz ( $\lambda = 0.3$  meters), then the maximum Doppler shift is  $f_d = v/\lambda = 100$  Hz. This illustrates that relative to the carrier frequency, the Doppler shift is typically quite small, but relative to baseband frequencies it can be relatively large. Also observe that for a constant vehicle velocity, the Doppler shift becomes larger as the carrier frequency increases.  $\square$

In addition to affecting the propagation distance and angle of incidence, the reflection in Figure 5-26 will also affect the attenuation constant and add an unknown phase shift due to the reflection coefficient.

The Doppler shift by itself might not be a big issue, since it results in an offset in a carrier frequency that might not be too precisely known in the first place. The more substantive effect occurs when there are two or more paths, each with different Doppler shifts because their incident angles at the receiver are different. If the delay spread of the different paths is small, we can assume a narrowband model; that is, the different delays of the arriving replicas of the baseband signal  $u(t)$  are insignificant for the baseband frequencies of interest. The resulting superposition of different Doppler shifts can result in a rapidly fluctuating phase and amplitude. For example, for a set of paths with amplitude  $A_i$ , delays  $\tau_i = \tau$  assumed to be the same on all paths (which is the narrowband model), phase shifts  $\phi_i$  at time zero, maximum Doppler shift



**Figure 5-27.** A model for the baseband channel with a receiver in motion at uniform velocity.



**Figure 5-28.** Amplitude and phase of  $r(t)$  resulting from the superposition of two paths with Doppler shift of 0 and 100 Hz, with  $A_1 = 1$  and  $A_2 = 0.9$ .

$\omega_d$ , and angles of incidence  $\theta_i$ , the receive complex baseband signal is

$$\sqrt{2} \cdot \text{Re} \left\{ \sum_i A_i u(t - \tau_i) e^{j(\phi_i - \omega_d \cos \theta_i t)} \right\} = \sqrt{2} \cdot \text{Re} \{ u(t - \tau) r(t) \}, \quad (5.65)$$

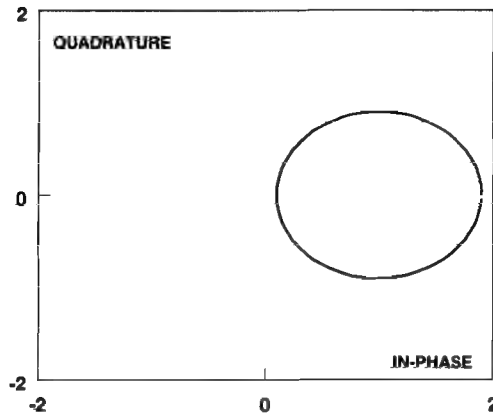
where

$$r(t) = \sum_i A_i e^{j(\phi_i - \omega_d \cos \theta_i t)}. \quad (5.66)$$

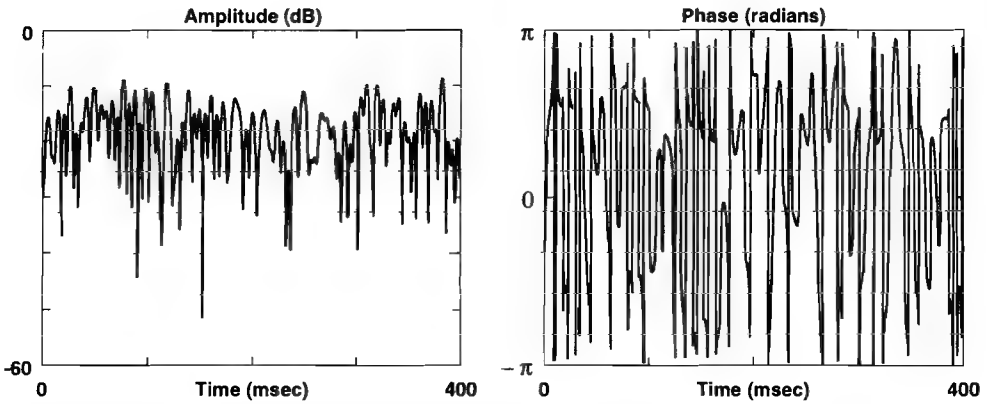
The basic equivalent baseband channel model is shown in Figure 5-27. The effect is multiplication by a complex-valued waveform  $r(t)$ . It is instructive to plot this waveform for a couple of cases. For example, we show in Figure 5-28 the effect of adding two carriers with a relative 100 Hz Doppler shift. The result is fades at 10 msec intervals, the 10 msec being the reciprocal of the relative Doppler shift. There are periodic very rapid phase jumps, corresponding precisely to the times at which there are large amplitude fades. This effect is explained by Figure 5-29, which shows a polar plot. The waveform  $r(t)$  follows the circular trajectory shown, where the angular velocity is constant. The amplitude fades occur when the trajectory comes near the origin, which coincides with time that the phase changes most rapidly.

These plots are repeated for 40 signals arriving from uniformly-spaced directions in Figure 5-30 and Figure 5-31. While the result is qualitatively similar, the trajectory is much more complicated and random-looking. Again there are occasional deep amplitude fades, which coincide with rapid phase variations. The timescale of these deep fades is again on the order of 10 msec, which is the reciprocal of the maximum Doppler frequency. This also corresponds to the time the receiver traverses a half of a wavelength at the carrier frequency.

The very chaotic change in amplitude and phase with time shown in Figure 5-30 can be characterized statistically employing the central limit theorem. Returning to the model of Figure 5-27, we can model the multiplicative signal  $r(t)$  as a random process  $R(t)$ . Examining this process at some point in time  $t_0$ , the phase of the  $i$ -th



**Figure 5-29.** Polar plot of the trajectory of  $r(t)$  for the same case as Figure 5-28.



**Figure 5-30.** a) The amplitude and b) phase for the superposition of 40 signals arriving at uniform angles, each with the same amplitude and random phases.

incident path is given by  $\xi_i = \phi_i - \omega_d \cos \theta_i t_0$ . Since the phases  $\phi_i$  are very sensitive functions of the initial position, it is reasonable to assume that the  $\xi_i$  are i.i.d. uniform random variables on the interval  $[0, 2\pi]$ . Writing the real and imaginary parts independently,

$$\operatorname{Re}\{R(t_0)\} = \sum_i A_i \cos \xi_i, \quad \operatorname{Im}\{R(t_0)\} = \sum_i A_i \sin \xi_i \quad (5.67)$$

where each term is the sum of independent random variables. By the central limit theorem, as the number of terms increases both  $\operatorname{Re}\{R(t_0)\}$  and  $\operatorname{Im}\{R(t_0)\}$  will be Gaussian distributed, and hence  $R(t)$  will be a complex-valued Gaussian random variable.

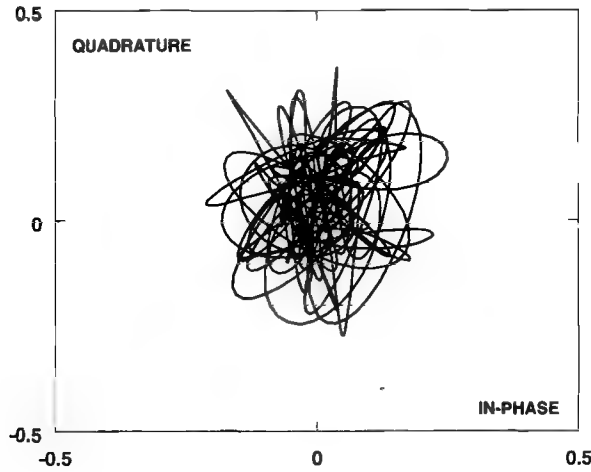


Figure 5-31. Polar plot for the same case as Figure 5-30.

**Exercise 5-5.**

Show that, with the assumption that the  $\xi_i$  are i.i.d. uniform random variables,

$$E[(\text{Re}\{R(t_0)\})^2] = E[(\text{Im}\{R(t_0)\})^2] = \sigma^2, \quad (5.68)$$

$$\sigma^2 = \frac{1}{2} \sum_i A_i^2, \quad (5.69)$$

$$E[\text{Re}\{R(t_0)\} \text{Im}\{R(t_0)\}] = 0. \quad (5.70)$$

□

When

$$R(t_0) = R e^{j\Theta} \quad (5.71)$$

is a complex-valued Gaussian random variable with identically-distributed and independent real and imaginary parts, then  $R$  is a Rayleigh-distributed random variable,

$$f_R(r) = \begin{cases} \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, & r \geq 0 \\ 0, & r < 0 \end{cases} \quad (5.72)$$

and the phase  $\Theta$  will be uniformly distributed on  $[0, 2\pi]$ . The amplitude  $R$  is the envelope of the received carrier, and  $\Theta$  is the phase, so we can say that the envelope has a Rayleigh distribution and the phase is uniform.

The conclusion is that when a CW carrier is transmitted, the received signal  $R(t)$  is well approximated as a complex Gaussian process. The power spectrum of that



process can be calculated, if we make assumptions about the distribution of arriving power vs. angle. This is because the frequency of an arriving component depends directly on the cosine of the angle of arrival. Let  $R(t)$  have power spectrum  $S_R(j\omega)$ . The contribution to  $R(t)$  arriving at angle  $\theta$  is at frequency  $(\omega_c + kv) \cdot \cos\theta$ . This implies that  $S_R(j\omega)$  is confined to frequency band  $[\omega_c - kv, \omega_c + kv]$ . In particular, the total power arriving in band  $[\omega_0, \omega_c + kv]$  corresponds to angle of arrivals in the range

$$kv \cdot \cos\theta + \omega_c \geq \omega_0, \quad \text{or } |\theta| \leq \theta_0 = \cos^{-1} \left[ \frac{\omega_0 - \omega_c}{kv} \right]. \quad (5.73)$$

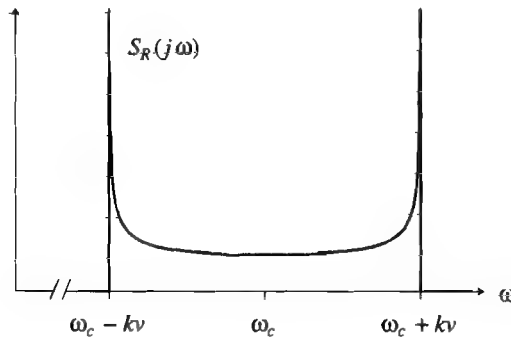
If we assume, for example, that a total received power  $P$  is arriving uniformly spread over all angles  $|\theta| \leq \pi$ , then the portion of the power arriving in band  $[\omega_0, \omega_c + kv]$  must be  $P \cdot \theta_0 / \pi$ . Thus,

$$\int_{\omega_0}^{\omega_c + kv} S_R(j\omega) \frac{d\omega}{2\pi} = \frac{P}{\pi} \cdot \cos^{-1} \left[ \frac{\omega_0 - \omega_c}{kv} \right], \quad (5.74)$$

and differentiating both sides with respect to  $\omega_0$ , the power spectrum is

$$S_R(j\omega) = \frac{2P}{\sqrt{(kv)^2 - (\omega - \omega_c)^2}}, \quad |\omega - \omega_c| \leq kv, \quad (5.75)$$

and zero elsewhere (of course the spectrum is symmetric about  $\omega = 0$ ). This power spectrum is plotted for positive frequencies in Figure 5-32, where we see that the power is concentrated in the region of frequencies  $\omega_c \pm kv$ . A sample function of a random process with this power spectrum will look like a random version of the deterministic signal  $\cos(\omega_c t) \cos(kvt)$ , since the latter has a Fourier transform that consists of delta functions at  $\omega_c \pm kv$ . This AM-DSB signal is the carrier multiplied by an envelope with periodic zero crossings (fades) spaced at  $\pi/kv = \lambda/2v$  sec intervals.



**Figure 5-32.** The Doppler power spectrum of the received carrier for a vehicle traveling at velocity  $v$ , assuming the received signal power is spread uniformly over all angles of arrival.

This is just the time it takes for the vehicle to travel a half wavelength. Thus, temporally, Rayleigh fading exhibits a strong tendency toward fades every half wavelength when the power is uniformly spread over all incoming angles.

**Example 5-26.**

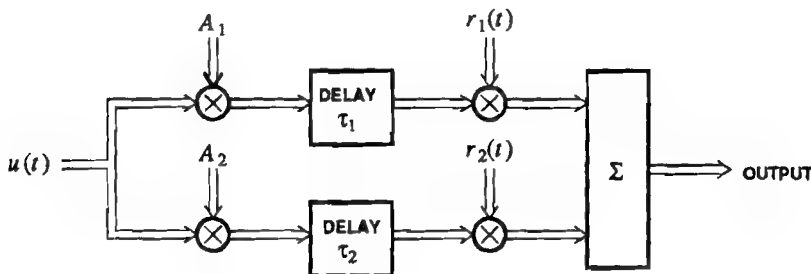
If the vehicle velocity is 100 km/hr and the carrier frequency is 1 GHz, the maximum Doppler frequency is approximately 100 Hz. This means that the individual paths coming into the receiver can have Doppler shifts on the order of  $\pm 100$  Hz, or the bandwidth of the passband signal is increased by approximately 200 Hz due to the motion of the vehicle. The wavelength is about 0.3 meters, so that the time it takes the vehicle to travel a half wavelength is

$$t = \frac{0.15 \text{ meters}}{30 \text{ meters/sec}} = 5 \text{ msec} . \quad (5.76)$$

We can expect significant fades approximately every 5 msec, which happens to be the reciprocal of the 200 Hz range of Doppler shifts.  $\square$

The model of Figure 5-27 and the Rayleigh fading derivation assumed a narrowband model; that is, the delay spread is small with respect to the reciprocal of the bandwidth, or equivalently that delays  $\tau_i$  in (5.65) are identical over all paths. Thus, the model must be modified to accommodate a wideband model, when the signal bandwidth is too large. Usually this is handled as follows. First, any given reflection, like off a high-rise building, is actually a complicated superposition of multiple reflections, where the delay spread across these reflections is small enough to obey the narrowband model. Thus, this single reflection can actually be represented by a narrowband Rayleigh fading model with an associated delay  $\tau_1$ . Now if there is a second reflection with a significantly different delay, it can be represented by another narrowband Rayleigh fading model with delay  $\tau_2 \neq \tau_1$ . The broadband model follows from superposition of these narrowband models.

A two-path broadband model is illustrated in Figure 5-33. The complex-baseband signal  $u(t)$  experiences the two path delays  $\tau_1$  and  $\tau_2$ , and the two delay outputs are multiplied by independent complex-Gaussian processes  $r_1(t)$  and  $r_2(t)$ . Each path also has an associated attenuation  $A_i$ , and a static phase shift which can be



**Figure 5-33.** A broadband two-path model, where each path is assumed to be independently Rayleigh fading.

subsumed in  $r_1(t)$  and  $r_2(t)$ . This broadband model is easily generalized to an arbitrary number of paths.

## 5.5. TELEPHONE CHANNELS

Most locations in the world can be reached over the public telephone network, so the voiceband channel is an almost universal vehicle for data communication. The design of digital modems for telephone channels is challenging, because the channel was designed primarily for voice, and impairments that are not serious for voice can be debilitating for data signals. The telephone channel is a prime example of a composite channel, consisting of many media such as wire pairs, satellite channels, coaxial cables, terrestrial microwave links, and optical fiber. Even more important than the media are the many modulation systems built on top of these media, such as pulse-code modulation and single-sideband modulation. The characteristics of the channel vary widely depending on the particular connection. It is useful to discuss these characteristics, not only because of the importance of this particular channel, but also because we encounter many impairments that occur in other situations as well.

### 5.5.1. Measured Performance of Telephone Channels

Because of the wide variety of possible connections, there is no simple analytical characterization of the telephone channel. Modem designers rely rather on statistical surveys of telephone circuits. In the U.S., a comprehensive survey was conducted in 1969-70 [22] and again in 1982-83 [23]. The data in this section comes primarily from interpretation of the second survey. A modem designer needs to determine the acceptable percentage of telephone connections over which the modem will perform, and then find the parameter thresholds that are met or exceeded by that percentage of channels. The resulting thresholds can be quite sensitive to the percentage.

#### Example 5-27.

According to the 1982-83 connection survey, 99% of end-to-end channels attenuate a 1004 Hz tone 27 dB or less. But 99.9% of channels attenuate the same tone 40 dB or less. To get the extra 0.9% coverage, an additional 13 dB of loss must be tolerated.  $\square$

In Table 5-1 we give typical worst-case figures assumed for some of the impairments on the channel. The percentage of telephone channels that exceed this performance is roughly 99%. Linear distortion is a major impairment that is missing from the table because it is difficult to summarize concisely. It is discussed below, followed by discussions of the remaining impairments.

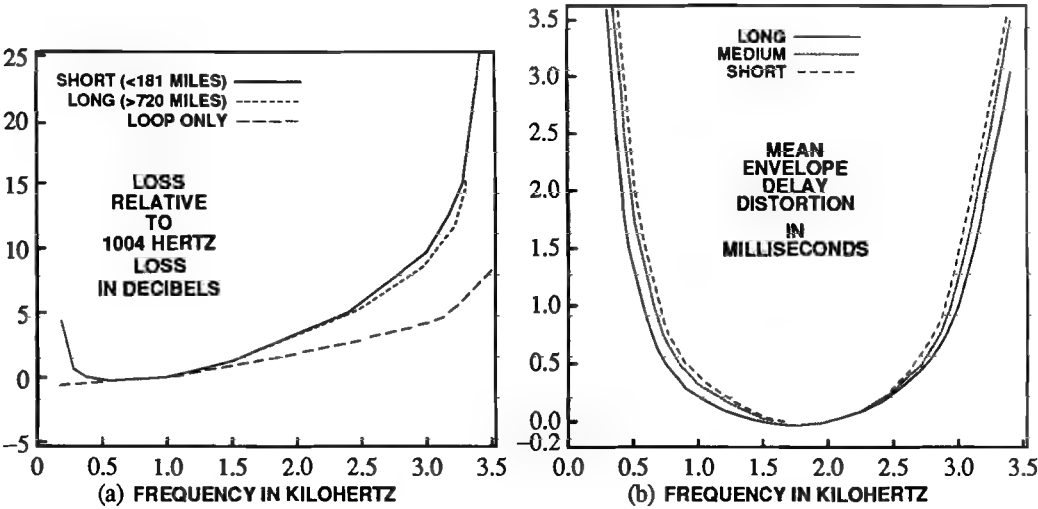
#### Linear Distortion

The frequency response of a telephone channel can be approximated by a linear transfer function  $B(j\omega)$ , roughly a bandpass filter from 300 to 3300 Hz. This bandwidth is chosen to give acceptable voice quality in the network, and is enforced by bandpass filters in analog and digital modulation systems used in the network. A typical transfer function of a telephone channel is illustrated in Figure 5-34, using

Impairment	level
Attenuation of a 1004 Hz tone	27 dB
Signal to C-notched noise ratio	20 dB
Signal to second harmonic distortion ratio	34 dB
Signal to third harmonic distortion ratio	33 dB
Frequency offset	3 Hz
Peak to peak phase jitter (2-300 Hz)	20 degrees
Peak to peak phase jitter (20-300 Hz)	13 degrees
Impulse noise (-4 dB threshold)	4 per minute
Phase hits (20 degree threshold)	1 per minute
Round trip delay (no satellites)	50 ms

**Table 5-1.** Typical worst-case impairments assumed for telephone channels. Roughly 99% of the telephone circuits measured in the 1982-83 connection survey [23] meet or exceed this performance.

traditional terminology that we now will explain. Amplitude distortion, the magnitude of the frequency response, is plotted as attenuation (or loss) vs. frequency. Amplitude distortion is often summarized as a set of *slope distortion* numbers, which attempt to capture images such as Figure 5-34a. A typical slope distortion measure is



**Figure 5-34.** The attenuation (a) and envelope delay distortion (b) of a typical telephone channel as a function of frequency. The attenuation is given relative to the attenuation of a 1004 Hz tone, and the envelope delay distortion relative to 1704 Hz, where it is near its minimum value [23].

the worst of two differences, (1) the loss at 404 Hz minus the loss at 1004 Hz and (2) the loss at 2804 Hz minus the loss at 1004 Hz. For 99% of telephone connections, that number is less than 9 dB. Several other slope distortion characterizations are found in the literature, but they are difficult to use in practice. We refer interested readers to the connection survey [23].

Interestingly, the attenuation in Figure 5-34a is almost precisely the typical attenuation of the local loop from the 1980 survey [24] combined with the typical frequency response of the filters in the PCM modulators in Figure 5-3, suggesting that these are the dominant sources of frequency-dependent attenuation.

Phase distortion, the deviation from linear of the phase response of  $B(j\omega)$ , is traditionally described as *envelope delay distortion*. *Envelope delay* is defined as the negative of the derivative of the phase of the received signal with respect to frequency, and hence measures the deviation from a linear phase response. Envelope delay distortion is often summarized by a set of numbers, much as the magnitude response is summarized by slope distortion. For details see [24].

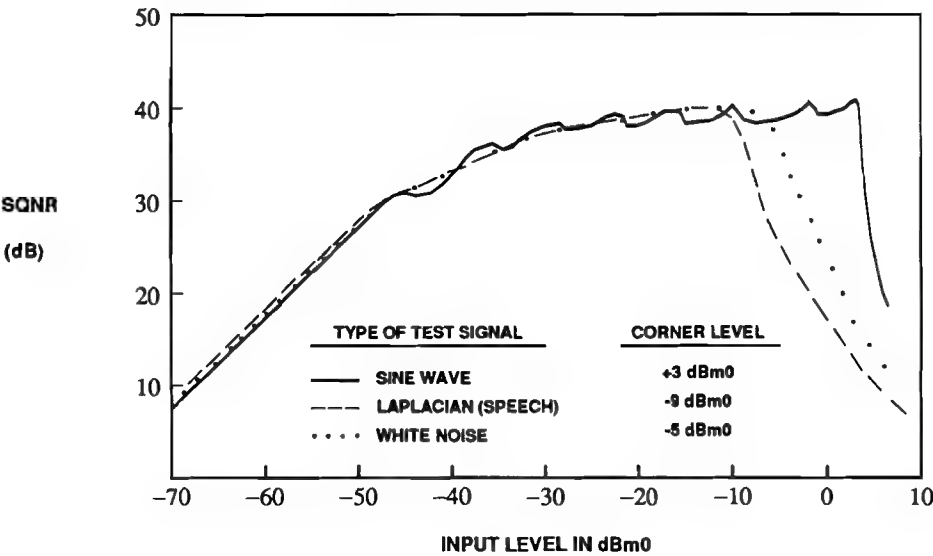
The overall attenuation of the channel is typically measured at 1004 Hz, where the attenuation is usually near its minimum. The attenuation is usually about 6 dB between one local switch and another, to which is added the loss of the local loops at each end.

## Noise Sources

In addition to attenuation, there is noise present on the voiceband channel, primarily from four sources: *quantization noise*, *thermal noise*, *crosstalk*, and *impulse noise*. We discuss them in order of increasing importance.

Crosstalk of the type discussed in Section 5.2 is one impairment that is more severe for voice than for data, so it has largely been eliminated from the network. Induction of interfering tones at 60 Hz and its harmonics (50 Hz in Europe) from power lines is more significant. As on other communication channels, thermal noise is an important impairment. *Impulse noise* consists of sudden, large spikes of short duration and is measured by counting the number of times the noise exceeds a given threshold. Impulse noise is due to electromechanical switches in the network, such as in telephone switches and dial telephones. Impulse noise is not well characterized, and modern designs are not heavily influenced by its presence.

The dominant source of noise is *quantization error* introduced by PCM systems, as in Figure 5-3. Quantization error is a consequence of using a limited number of bits to represent each sample in the PCM system. While the quantization error is deterministically dependent on the signal, the randomness of the signal usually gives quantization error a "noise-like" characteristic. It has an approximately white power spectrum, and the level of noise is usually measured by the *signal-to-quantization-noise ratio (SQNR)*. The SQNR for a single quantizer as encountered in the U.S. telephone network is illustrated in Figure 5-35. Note that over an input range of about 30 dB (-40 to -10 dBm0) the SQNR varies by only about 6 dB (33 to 39 dB). This relatively constant SQNR implies that the quantization error power varies almost in direct proportion to the signal power; that is, it is not constant independent of the signal as for thermal noise. A thorough study of this noise is given in [25]. For the fastest



**Figure 5-35.** SQNR as a function of the absolute input signal power for three different types of inputs. The Gaussian and Laplacian inputs are random, with the latter approximating the p.d.f. of speech samples [23].

voiceband data modems, it can be the dominant impairment. For lower speed modems, it is adequately approximated by white Gaussian noise.

In Table 5-1, the noise is labeled *C-notched noise*, which refers to a particular filter applied to the noise prior to measurement of power. This filter is chosen on the basis of subjective effects for voice, and if the noise is white has no effect beyond a fixed offset in the measured power. Quantization error power is measured by applying a *holding tone*, usually at 1004 Hz, and filtering out this tone with a deep (-50 dB) notch filter prior to the measurement of the remaining quantization error with a C-message weighted filter.

**Nonlinear Distortion**

Nonlinear distortion is due to imperfections in amplifiers and also to tracking errors between A/D and D/A converters. Because of its relatively low level, nonlinear distortion is a significant impairment only for the most elaborate, highest data-rate modems.

**Frequency Offset**

Frequency offset is peculiar to telephone channels and channels with Doppler shift. If the input to the channel is  $x(t)$ , with Fourier transform  $X(j\omega)$ , and the channel has a frequency offset of  $\omega_0$  radians, and no other impairments, then the output of the channel has Fourier transform

$$Y(j\omega) = \begin{cases} X(j\omega - j\omega_0) & \text{for } \omega > 0 \\ X(j\omega + j\omega_0) & \text{for } \omega < 0 \end{cases} \quad (5.77)$$

This small shift in the spectrum of signal has important implications for carrier recovery (Chapter 16) and echo cancellation (Chapter 19).

#### Exercise 5-6.

We saw in Chapter 2 that passband data signals can be expressed in the form

$$x(t) = \text{Re} \{ s(t) e^{j\omega_c t} \}, \quad (5.78)$$

where  $s(t)$  is a complex-valued baseband data signal and  $\omega_c$  is the carrier frequency. Show that the effect of a frequency offset on the channel is a received signal

$$y(t) = \text{Re} \{ s(t) e^{j(\omega_c - \omega_0)t} \}. \quad (5.79)$$

In effect, the carrier frequency has been shifted by  $\omega_0$ . Assume that  $\omega_0 > 0$  and  $s(t)$  is bandlimited so that  $S(j\omega) = 0$  for  $|\omega| > \omega_c$ .  $\square$

Frequency offset is a consequence of using slightly different frequencies to modulate and demodulate single-sideband (SSB) signals in analog transmission facilities (Figure 5-1). It is allowed because it has no perceptible effect on speech quality, and can be compensated by proper design in voiceband data modems.

### Phase Jitter

Phase jitter on telephone channels is primarily a consequence of the sensitivity of oscillators used for carrier generation in SSB systems (Figure 5-1) to fluctuations in power supply voltages. Since power supply fluctuations are often at 60 Hz or harmonics thereof, the largest components of phase jitter are often at these frequencies. Phase jitter is measured by observing the deviation of the zero crossings of a 1004 Hz tone from their nominal position in time.

Phase jitter can be viewed as a generalization of frequency offset. If the phase jitter on a channel is  $\theta(t)$ , the effect on the transmitted signal of (5.78) is a received signal of the form

$$y(t) = \text{Re} \{ s(t) e^{j(\omega_c t + \theta(t))} \}. \quad (5.80)$$

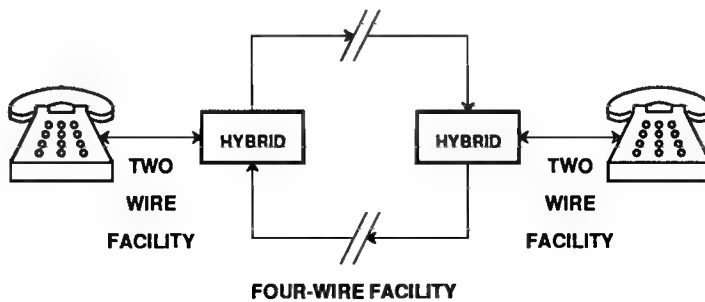
A phase jitter of  $\theta(t) = \omega_0 t$ , amounts to frequency offset. It is common for  $\theta(t)$  to have oscillatory components at the power line frequency (50 or 60 Hz) and harmonics. If we simply demodulate this signal using the carrier  $e^{j\omega_c t}$ , we recover a distorted baseband signal  $s(t) e^{j\theta(t)}$  rather than the desired  $s(t)$ . To mitigate this distortion, it is common in carrier recovery (Chapter 16) to include algorithms designed to track and remove this undesired phase jitter.

A phase hit is an abrupt change in the nominal phase of a received sinusoidal signal lasting at least 4 ms. There is little that can be done to defend a modem against this degradation, but it must be taken into account in the design of the carrier recovery (Chapter 16).

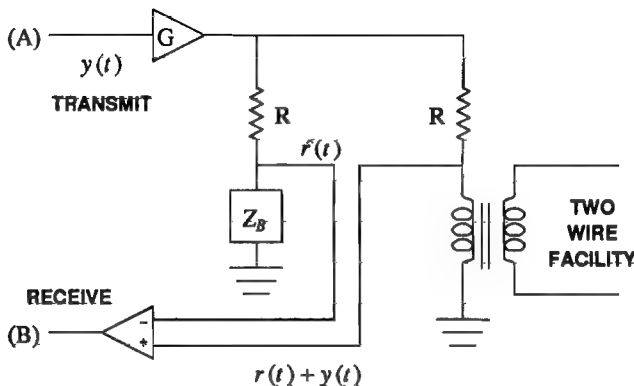
## Delay and Echo

*Delay* and *echo* are the final impairments in telephone channels that we will consider. A simplified telephone channel is shown in Figure 5-36. The *local loop*, which is the twisted wire pair connecting the central office with customer premise, is used for transmission in both directions. Both signals share the same wire pair. At the central office, a circuit called a *hybrid* separates the two directions of transmission. Longer distance facilities are *four-wire*, meaning that the two directions of transmission are physically separated.

One possible implementation of the hybrid circuit is shown in Figure 5-37. The signal from the other end of the two-wire facility is fed through to the receive port. The transmit signal appears at the transformer as a voltage divider with impedances  $R$  and  $Z_0$ , where the latter is the input impedance of the two-wire facility. We cancel



**Figure 5-36.** A simplified telephone channel, showing the two-wire local loop and the four-wire transmission facility.



**Figure 5-37.** An electronic hybrid. To avoid leakage of the receive signal (A) into the transmit path (B) the impedance  $Z_B$  should exactly match the impedance of the transformer and two-wire facility.



this undesired feedthrough by constructing another voltage divider with a *balance* impedance  $Z_B$ . When  $Z_B = Z_0$ , the loss from transmit to receive port is infinite. In practice, a fixed compromise impedance  $Z_B$  is used, and a component of the receive signal (A) can leak through to (B) with an attenuation as small as 6 to 10 dB due to the variation in impedance of the two-wire facility.

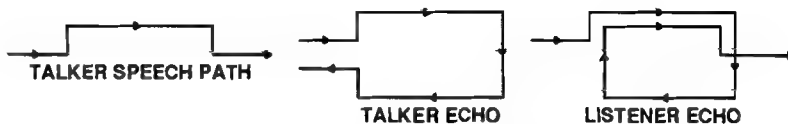
The signal and two types of echo paths for the configuration of Figure 5-36 are shown in Figure 5-38. An *echo* is defined as a signal component that has taken any path other than the *talker speech path*. The *talker echo* is the signal that leaks through the far-end hybrid and returns to the sender (talker). The *listener echo* is the component of the talker echo that leaks through the near-end hybrid and returns again to the listener. This echo is similar to multipath propagation on radio channels (Section 5.4). The length of the telephone channel determines the round-trip echo delay. Echoes from the near end of the connection typically undergo zero to 2 msec of delay, whereas far-end echoes can have round-trip delays of 10-60 msec for terrestrial facilities, or up to 600 msec on satellite connections.

To mitigate the effects of echo on speech quality, several strategies co-exist on the network. The effect of each strategy on data signals is different. For short delays, loss is added in the talker speech path, which is advantageous because the echoes experience this loss more than once. This loss, plus the loss of the subscriber loops at each end, is the source of the attenuation that must be accommodated by data transmission; it can be as high as 40 dB (at 1004 Hz). For longer delays, devices known as *echo suppressors* and *echo cancelers* are added to the connection.

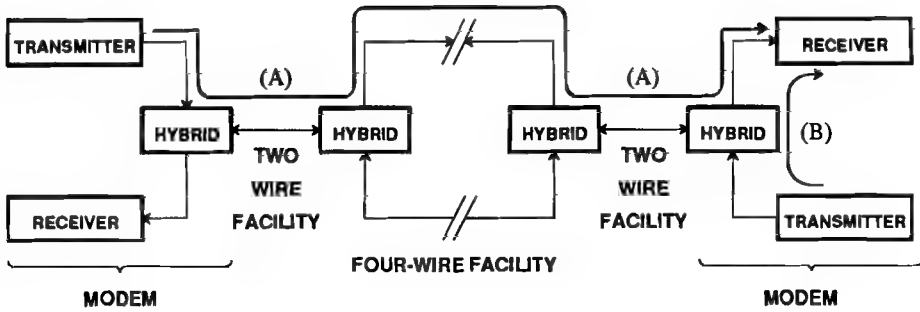
A *full-duplex* (FDX) modem is one that transmits and receives on the same telephone channel. Such a modem requires an internal two-to-four-wire conversion, as shown in Figure 5-39. Because of imperfect balance impedances of the hybrids, some of the transmitted signal echoes into the receiver and interferes with the weaker data signal from the far end. The hybrid echo loss may be as low as about 6 dB, and the received signal may have experienced as much as 40 dB loss, so the desired far-end signal may be as much as about 34 dB *below* the echo. Ways of dealing with this problem are discussed in Chapters 18 and 20.

### 5.5.2. Channel Capacity Compared to Practical Modems

Rough estimates of the capacity of a voiceband telephone channel indicate it is over 30,000 b/s. In Table 5-2 we summarize the bit rates achieved by existing standardized voiceband data modems. Bit rates as high as 28,800 b/s are envisioned,



**Figure 5-38.** Three of many possible signal paths in a simplified telephone channel with a single two-to-four-wire conversion at each end.



**Figure 5-39.** Two modems connected over a single simplified telephone channel. The receiver on the right must be able to distinguish the desired signal (A) from the signal leaked by its own transmitter (B).

although the higher rates may be achievable on a smaller fraction of possible connections. Indeed, several of the higher speed modems are used exclusively with *leased lines*, which can be conditioned for guaranteed quality.

## 5.6. MAGNETIC RECORDING CHANNELS

Digital communication is used not only for communication over a distance (from here to there), but also for communication over time (from now to then). The latter application is called *digital storage* or *recording*, and is usually accomplished using a magnetic medium in the form of a tape or disk. More recently, particularly in the context of read-only applications, optical storage media have been used as well.

### Example 5-28.

The compact disk ROM, an offshoot of a similar consumer audio technology, allows 600 megabytes of data to be stored on a single plastic disk 12 cm in diameter [26]. The bits are stored as small pits in the surface, and are read by spinning the disk, shining a laser diode on the surface, and detecting the reflected light with an optical pickup. □

Digital recording is of course used extensively in computing systems, but is increasingly used in addition for the storage of music [27,28] or voice.

### Example 5-29.

The compact disk digital audio system, which is a great commercial success, records music digitally using a similar technology to the compact disk ROM. The music is converted to digital using 16 bits per sample at a sampling rate of 44.1 kHz for each of two channels, for a total bit rate of about 1.4 Mb/s. Up to 70 minutes of material can be recorded on a single disk. □

speed	symbol	duplex	CCITT	modulation
(b/s)	rate	(method)	std.	
≤ 300	≤ 300	full(FDM)	V.21	2-FSK
1200	1200	half	V.23	2-FSK
1200	600	full(FDM)	V.22	4-PSK
2400	1200	half	V.26	4-PSK
2400	600	full(FDM)	V.22bis	16-QAM
2400	1200	full(EC)	V.26ter	4-PSK
4800	1600	half	V.27	8-PSK
4800	2400	full(EC)	V.32	4-QPSK
9600	2400	half	V.29	16-AM/PM
9600	2400	full(EC)	V.32	32-QAM+TC
14,400	2400	full(EC)	V.32bis	128-QAM+TC
≤ 28,800	≤ 3429	full(EC)	V.fast(V.34)	1024-QAM+TC

**Table 5-2.** Important standardized voiceband data modems are summarized here. The "duplex" column indicates whether a single channel is shared for both directions of transmission (full) or separate channels must be used for each direction (half). For full duplex modems, it also indicates whether frequency division multiplexing (FDM) or echo cancellation (EC) is used for multiple access (Chapter 18). The "CCITT std" column identifies the international standard that applies to this type of transmission. Finally, the "modulation" column identifies the type of modulation, which are discussed in Chapters 6 and 14. The numbers indicate the number of symbols in the alphabet. The "TC" in the V.32 and V.33 refers to trellis coding (Chapter 14).

**Example 5-30.**

Digital storage on disk drives is used in speech store-and-forward systems, which are essentially the functional replacement for telephone answering systems, except that they serve a number of customers. □

Digital recording offers some of the same advantages over analog recording as we discussed earlier for transmission. The principle advantage again is the *regenerative effect*, in which the recording does not deteriorate with time (except for the introduction of random errors which can be eliminated by coding techniques) or with multiple recordings and re-recordings. An additional advantage is the compatibility of digital recording with digital signal processing, which offers very powerful capabilities.

Magnetic tape or disk can be considered as a transmission medium in much the same manner as other media such as wires and fibers [29,30]. We will now briefly discuss the properties of that medium.

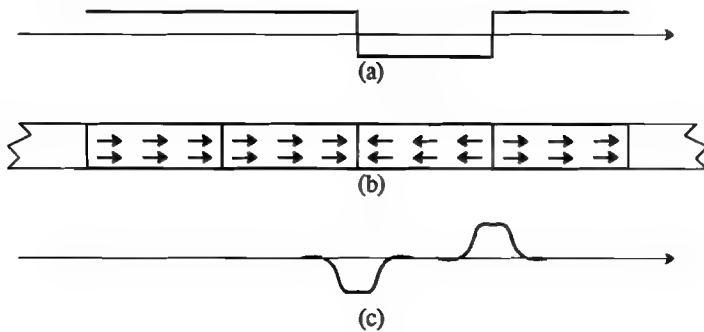
### 5.6.1. Writing and Reading

In the writing process, a magnetic field is generated in an electromagnet called a *head* as it passes at high speed over a ferric oxide magnetic medium, thereby orienting the direction of magnetization along a track in a nearby magnetic medium on the disk or tape [31]. On reading, when the oriented magnetic pattern passes under that same

head, it produces a voltage that can be sensed and amplified.

There are two basic types of recording. *Saturation recording* is almost always used for digital recording, in which the magnetization is saturated in one direction or the other. Thus, in saturation recording, the medium is constrained to be used for binary transmission; that is, only two levels are allowed. This is in contrast to wire and coaxial media in which multi-level transmission can be considered. The other form of magnetic recording is *a.c. bias recording*, in which the signal is accompanied by a much larger and higher frequency bias sinusoid for the purpose of linearizing the channel. A.c. bias recording is necessarily used in analog recording, where linearity is important, but has not been applied to digital recording because of the deterioration in signal-to-noise ratio and the fact that saturation recording is appropriate for binary modulation and demodulation techniques.

The magnetic recording process is qualitatively illustrated in Figure 5-40. For saturation recording, the voltage applied to the write head assumes one positive and one negative value corresponding to the two directions of desired magnetization. In Figure 5-40a it is assumed that a square wave corresponding to the binary sequence "1101" is applied to the write head. This waveform correspondence to a bit sequence is called *non-return to zero*, or *NRZ*. The bit stream is recorded on linear (tape) or circular (disk) *tracks* on the magnetic medium, and one track is shown in Figure 5-40b. Note the two directions of magnetization, schematically indicated by the arrows. The voltage on the read head (which is physically the same as the write head) during a read operation is shown in Figure 5-40c. As long as the magnetization is constant, no voltage is induced in the read head coil, but upon a *change* in magnetization there is a voltage induced (recall that the voltage induced in a coil is proportional to the *derivative* of the magnetic field). The polarity of that voltage is determined by the *direction of change* in magnetization.



**Figure 5-40.** Illustration of magnetic recording. a. The NRZ waveform applied to the record head corresponding to bit sequence "1101". The abscissa is time, but this is proportional to distance on the medium for constant velocity of the head. b. The magnetization of one track after saturation recording. b. The voltage on the read head coil corresponding to position of the read head, which at constant velocity is the same as time.

This write-read magnetic recording process can be viewed as a communication channel if we observe only the input and output voltage waveforms in Figure 5-40a and Figure 5-40c and ignore the physical medium of Figure 5-40b. Both of these waveforms represent signals in time, just like in communications, although there is a conceptually insignificant and indeterminate time delay between the write and read operations. Viewed as a communication channel, we see that the magnetic recording channel of Figure 5-40 inherently includes a differentiation operation. Another way of looking at this is that the channel is sensitive to only the *transitions* in the input waveform rather than its polarity. Therefore, from a digital communication point of view, we want to map the input bits into transitions in the input waveform rather than absolute polarity. The way in which this can be done will be considered in Chapter 6.

### 5.6.2. Linearity of the Magnetic Channel

The magnetic channel can be made linear in a special sense to be specified now. This linearity is a very desirable feature, in that it will greatly simplify system design.

The view of Figure 5-40 is oversimplified in that it assumes that the magnetization is in either one direction or the other. In fact, the tape medium contains a multiplicity of tiny magnetic particles, and each particle must indeed be magnetized in one direction or the other. The total net magnetization can assume almost a continuum of values, depending on the number of particles magnetized in each direction. Unfortunately this continuum of magnetization depends nonlinearly on the applied magnetic field, and displays hysteresis, and therefore the write process is highly nonlinear. On the other hand, the read process is very linear, in that the voltage induced on the read head is a linear function of the magnetization.

If the applied field to the recording head is strong enough and held long enough so that the medium is fully saturated, then the output of the read head displays a form of superposition. This is because this saturation destroys the memory of the hysteresis. This form of superposition is illustrated in Figure 5-41. If the response to a positive transition at time  $t$  is  $h(t)$ , and the response to a negative transition at time  $t + \Delta$  is  $-h(t + \Delta)$ , then the response to the positive followed by negative transition obeys superposition, and is

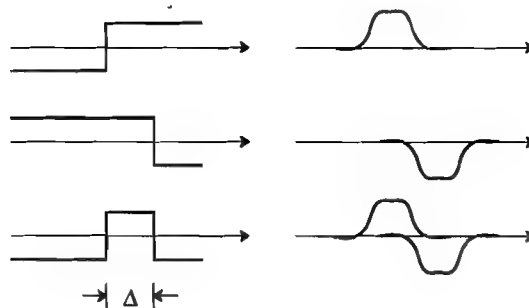


Figure 5-41. Superposition in the reading process of magnetic recording.

$$h(t) - h(t + \Delta) . \quad (5.81)$$

This is true with great accuracy *as long as* the time between transitions  $\Delta$  is larger than some threshold  $\Delta_0$ . This threshold is determined by the time to achieve full saturation of the medium in one direction or the other since the last transition, and depends on the design of the write head as well as the medium.

### 5.6.3. Noise on the Magnetic Channel

The noise impairments are very complicated on the magnetic channel, consisting of additive and multiplicative noise components. A major source of noise is due to the granularity of the medium. The total response of the head is the superposition of the responses to a multiplicity of magnetic particles. This discrete nature of the signal is similar to the quantum nature of the optical detection process (Section 5.3) with one important distinction. In optical detection we have only photons (or photoelectrons) or the absence of same, whereas in magnetics there are both positively and negatively magnetized particles. Thus, in optical detection, when there is no signal incoming there is also no quantum noise (neglecting dark current). In the magnetic reading process the particles are present whether or not there is a signal, or putting it another way the absence of a signal is represented by an equal number of positively and negatively magnetized particles. Hence, the *granular noise* in magnetic recording is present independent of the signal, and is therefore truly an additive noise phenomenon. Its spectrum is not white because it is filtered by the read head response, and in fact its power spectrum tends to be similar to the spectrum of the read signal.

*Zero crossing jitter* results from variations in the composition of the medium and the distance between the write head and the medium. The effect is a small jitter in the position of the read pulses. Another phenomenon is *amplitude modulation* of the received signal, a multiplicative noise phenomenon due to medium density fluctuations. An extreme form of amplitude modulation is the tape dropout, in which the signal level gets too small for reliable detection. Since dropouts are a reproducible effect, depending on position on the medium, they are often flagged in disk files so that these areas are not used for recording. Another phenomenon is interference from adjacent tracks, which is similar to the crosstalk experienced in multi-wire-pair cables and the interference between radio systems.

## 5.7. FURTHER READING

The literature on communication media is vast and scattered. We offer here some suggestions that may help the reader get started. The basic theory of transmission lines is covered in [1]. There are a number of books devoted to optical fiber devices [32,6,33,34,35,36,37,38] and a smaller number that concentrate on the noise and system issues of primary interest here [39,13,14]. A special issue of *IEEE Journal on Selected Areas in Communications* (November, 1984) is devoted to undersea lightwave communication. It contains numerous useful articles describing the special problems that arise in this hostile environment. Another issue (November, 1985) covers fiber optics for local communication, and concentrates on networking issues. Yet

another issue (December, 1986) is devoted to terrestrial fiber optics, and includes papers on reliability, economics, and networking issues. Finally, the Optical Society of America's *Journal of Lightwave Technology* is a good source of information.

There are available books on satellite [18] and mobile radio [21] design. Special issues of the *IEEE Journal on Selected Areas in Communications* in July 1984, June 1987, and January 1989 are devoted to mobile radio communication. Many of the papers propose modulation schemes that are robust in the presence of multipath fading. More specifically directed to multipath fading channels is another special issue (February 1987). Another issue (April 1987) is devoted to point-to-point digital radio, and yet another (January 1985) to broadcasting satellites.

Further information about characteristics of the telephone channel is best obtained by going directly to the published line surveys [22,23,24]. Special issues of *IEEE Journal on Selected Areas in Communications* (September 1984, and August and December 1989) are devoted to modulation and coding for the telephone channel.

A special issue of *IEEE Journal on Selected Areas in Communications* in January 1992 is devoted to recent results on magnetic recording channels.

## PROBLEMS

- 5-1. Show that for a terminated transmission line with real-valued characteristic impedance, the maximum power to the load is obtained in Figure 5-6b when  $Z_S = Z_L = Z_0$ .

- 5-2. For a transmission line, derive the relation

$$\lambda f = v \quad (5.82)$$

where  $f$  is the frequency of a propagating wave in Hz,  $\lambda$  is the wavelength in meters, and  $v$  is the velocity of the wave in meters/sec.

- 5-3. In subscriber loop wire-pair cables, it is common in some countries to have *bridged taps*, which are *open circuited* wire-pairs bridged in parallel on the main pair. Assume that a source has impedance equal to the wire-pair characteristic impedance, the wire-pair is terminated at the other end by its characteristic impedance, and that the wire-pair has a single bridged tap. Let the distance from source to tap be  $L_1$ , from tap to termination  $L_2$ , and let the length of the bridged tap be  $L_3$ .
- Find an expression for the transfer function of the wire-pair including bridged tap. Be sure to take advantage of the simplifications due to the terminations with the characteristic impedance.
  - Show that when the bridged tap is very long, it causes a fixed attenuation at all frequencies. What is that attenuation?
  - State intuitively what you would expect the response at the termination to be to a single transmitted pulse as a function of the length of the bridged tap.
  - Discuss what happens intuitively when the bridged tap approaches zero length.
- 5-4. Use Snell's law to show that in Figure 5-12 a ray will be captured by the fiber as long as the incident angle obeys

$$\sin(\theta_1) < (n_1^2 - n_2^2)^{1/2}. \quad (5.83)$$

This confirms that rays incident at small angles are captured, and those at larger angles are not.

5-5. Let the length of the fiber be  $L$ .

- (a) Show that the path length for a ray is equal to  $L \sec(\theta_2)$ .
- (b) Show that the path length varies from  $L$  to  $n_1^2 L / n_2^2$ . Thus, the larger the difference in index of refraction of core to cladding, the larger the range of captured angles, but also the larger the variation in the transit time of rays through the length of the fiber.

5-6. Assuming that the chromatic dispersion in a single mode fiber is 0.15 psec/km-GHz, evaluate numerically (5.30). Sketch the curve of repeater spacing vs. bit rate in the range of repeater spacings between 1 and 1000 km as limited by dispersion.

5-7. In an optical fiber receiver, assume the received optical power is  $P$ , the bit rate is  $R$  bits/sec.

- (a) Find the number of received photons per bit.
- (b) Show that for a constant number of photons per bit, the required received optical power is proportional to the bit rate.
- (c) Find the received optical power necessary to receive 100 photons per bit at a wavelength of 1.5  $\mu$ m and a bit rate of 1 Gb/s.
- (d) For the same conditions as c., assume you can launch one mwatt power into the fiber, and that the fiber loss at that wavelength is 0.2 dB per km. What is the distance that we can transmit?

5-8. A typical direct detection optical receiver requires about  $N = 2000$  photons per bit in the notation of Problem 5-7.

- (a) Derive the following formula [9] for the required received power at an optical detector at a wavelength of 1.5  $\mu$ m for this value of  $N$ ,

$$P_{dBm} = -65.8 + \log_{10} R_{Mb} \quad (5.84)$$

where  $P_{dBm}$  is the received power in dBm required and  $R_{Mb}$  is the bit rate in Mb/s. Note how the required power increases as the bit rate increases. In particular, each order of magnitude increase in bit rate increases the required power by only one dB.

- (b) Assuming 0 dBm launched power into the fiber, and 0.2 dB per km loss in the fiber, what is the allowable distance between repeaters at bit rates of 100 and 1000 Mb/s? You can assume that loss is the dominant impairment limiting repeater spacing.

5-9. Change the assumptions in Problem 5-8 to those that might better reflect fundamental limits [9]: A launched signal power of 20 dBm and 20 photons per bit required at the receiver.

5-10. Suppose we have a system requirement that a total bit rate of  $R_T$  must be transmitted over a distance of  $L_T$  using a set of parallel repeatered transmission lines (wire cable or fiber). In each repeater span we have as design parameters the bit rate  $B$  and repeater spacing  $L$ . Show that the total number of repeaters is minimized when the quantity  $B \cdot L$  is maximized for the given transmission technology. Thus, if the repeaters are the dominant transmission cost, we want to maximize the product of the bit rate and the distance for each technology.

5-11.

- (a) Derive the following relationship between repeater spacing and bit rate, using the assumptions of Problem 5-8, and assuming a fiber loss of  $\gamma_0$  dB/km:

$$L = \frac{65.8 - \log_{10} R_{Mb}}{\gamma_0} \quad (5.85)$$

You can assume that the number of received photons per bit is held constant and the transmit power is held constant at 0 dBm.

- (b) Sketch this relation for the range of bit rates between 1 Mb/s and 10,000 Mb/s and a fiber loss of 0.2 dB/km and verify that Figure 5-16 is qualitatively correct in predicting this loss-limited region.
- (c) Using the results of Problem 5-10, argue that it will be best to increase the bit rate until dispersion becomes the controlling impairment, if the criterion is to minimize the number of



repeaters.

- 5-12. The available thermal noise power in a bandwidth  $B$  Hz is  $kT_n B$ . For a resistor generating thermal noise, the noise source can be modeled as either a series voltage source or a parallel current source. Show that the voltage source has mean-squared voltage  $4kT_n RB$  and the current source has mean-squared current  $4kT_n B/R$  within bandwidth  $B$  for a resistance of  $R$  ohms.
- 5-13. At 6 GHz, what is the diameter of a circular aperture antenna that has an antenna gain of 40 dB with a 70% efficiency?
- 5-14. A radio channel with bandwidth 30 MHz is centered at 6 GHz. What is the difference in decibels in the path loss between the high and low end of this channel? At which end is the loss the minimum? (CAUTION: The antenna gains are a function of frequency.)
- 5-15. Compare the tradeoff between repeater spacing  $d$  and transmitted power  $P_T$ , assuming that the received power  $P_R$  is held constant for the following two media:
- (a) Metallic cable or fiber optics with loss  $\gamma_0$  dB per km.
  - (b) A microwave radio system.
  - (c) For which medium does the transmitted power have the most impact?
- 5-16. Develop the following formula which relates the free-space loss between isotropic radiators in dB, the distance  $d_{\text{km}}$  between radiators in km, and the frequency  $f_{\text{GHz}}$  in GHz,
- $$\text{Loss(dB)} = 92.4 + 20\log_{10} d_{\text{km}} + 20\log_{10} f_{\text{GHz}} \quad (5.86)$$
- Note the dependence of this loss on distance and frequency.
- 5-17. In this problem we will determine how to combine noise sources in a radio receiver with different noise temperatures to yield a single equivalent noise source. For the configuration of Figure 5-42 where the noise temperature of the three noise sources  $n_i(t)$  are  $T_i$ , find the relationship between these noise temperatures such that the two systems will have the same SNR. The parameter  $G$  is the power gain of an amplifier, i.e., the ratio of input to output power.
- 5-18. Use the results of Problem 5-17 to find the equivalent noise temperature at the input to the

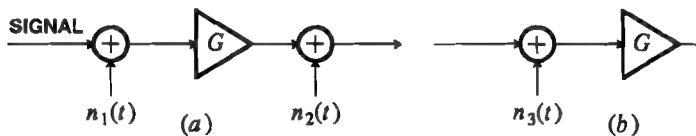


Figure 5-42. Illustration of the combination of two noise sources into a single equivalent noise source referenced to the input of the system.

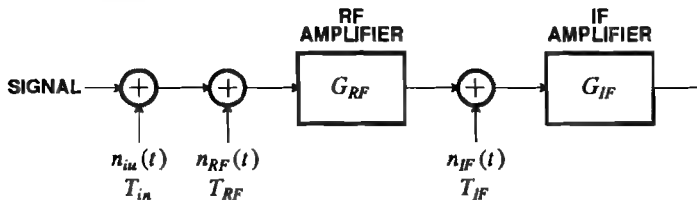


Figure 5-43. Several noise sources introduced at the input and internally to a receiver.

receiver of Figure 5-43, where each of the circuit elements is assumed to be noiseless with an associated noise source with associated noise temperature.

- 5-19. Estimate the delay spread for the two-path ground-reflection model of Exercise 5-3 for a spacing of antennas by 3 km and antenna height of 50 meters. What is the approximate baseband bandwidth over which the narrowband model is applicable?
- 5-20. Suppose the incoming power in a Rayleigh fading scenario does not arrive at a moving vehicle uniformly spread over all angles. Describe qualitatively how you would expect the power spectrum of Figure 5-32 to be affected under the following conditions:
- The vehicle is driving toward the transmitter, and more power is arriving from the direction of the transmitter than other directions.
  - A lot of power is reflecting off a nearby mountain, so that more power is arriving at the vehicle from the side (relative to the direction of motion) than any other direction.
- 5-21. Consider a SSB analog voice transmission system embedded in the telephone network. Suppose that the carrier frequency  $f_c$  is nominally 1 MHz. In practice, the transmitter and receiver will be designed with components that yield modulating and demodulating frequencies that are slightly off. Component manufacturers often express the accuracy of precision parts in *parts per million*, instead of percent (which is *parts per hundred*). How accurate (in parts per million) do the modulating and demodulating oscillator frequencies have to be to guarantee less than 3 Hz frequency offset?
- 5-22. Suppose that a data signal

$$x(t) = \text{Re}\{s(t)e^{j\omega_0 t}\} \quad (5.87)$$

is transmitted over a telephone channel with frequency offset  $\omega_0$  and sinusoidal phase jitter with frequency  $\omega_p$  and amplitude  $a$ . Assume there are no other impairments. Give an expression for the received signal.

## REFERENCES

- G. C. Temes and J. W. LaPatra, *Introduction to Circuit Synthesis and Design*, McGraw-Hill, New York (1967).
- Bell Laboratories Members of Technical Staff, *Transmission Systems for Communications*, Western Electric Co., Winston-Salem N.C. (1970).
- P. Bylanski and D. G. W. Ingram, *Digital Transmission Systems*, Peter Peregrinus Ltd., Stevenage England (1976).
- S. V. Ahamed, P. P. Bohn, and N. L. Gottfried, "A Tutorial on Two-Wire Digital Transmission in the Loop Plant," *IEEE Trans. on Communications* COM-29(Nov. 1981).
- K. C. Kao and G. A. Hockham, "Dielectric-Fiber Surface Waveguides for Optical Frequencies," *Proc. IEE* 113 p. 1151 (July 1966).
- D. B. Keck, "Fundamentals of Optical Waveguide Fibers," *IEEE Communications* 23(5)(May 1985).
- J. T. Verdeyen, *Laser Electronics*, Prentice Hall, Englewood Cliffs N.J. (1981).
- D. B. Keck, "Single-Mode Fibers Outperform Multimode Cables," *IEEE Spectrum* 20(3) p. 30 (March 1983).
- J. E. Midwinter, "Performance Boundaries for Optical Fibre Systems," *NATO Advanced Study Institute*, (July 1986).

10. P. S. Henry, "Introduction to Lightwave Transmission," *IEEE Communications* 23(5)(May 1985).
11. T. Li, "Structures, Parameters, and Transmission Properties of Optical Fibers," *Proc. IEEE* 68(10) p. 1175 (Oct. 1980).
12. S. R. Forrest, "Optical Detectors: Three Contenders," *IEEE Spectrum* 23(5) p. 76 (May 1986).
13. S. D. Personick, *Optical Fiber Transmission Systems*, Plenum Press, New York (1981).
14. S. D. Personick, *Fiber Optics Technology and Applications*, Plenum Press, New York (1985).
15. D. Taylor and P. Hartmann, "Telecommunications by Microwave Digital Radio," *IEEE Communications Magazine* 24(8) p. 11 (Aug. 1986).
16. J. Mikulski, "DynaT\*A\*C Cellular Portable Radiotelephone System Experience in the U.S. and the U.K.," *IEEE Communications Mag.* 24(2) p. 40 (Feb. 1986).
17. V. MacDonald, "The Cellular Concept," *BSTJ* 58(1)(Jan. 1979).
18. T. Pratt and C. W. Bostian, *Satellite Communications*, John Wiley, New York (1986).
19. W. Rummier, R. Coutts, and M. Liniger, "Multipath Fading Channel Models for Microwave Digital Radio," *IEEE Communications Mag.*, (11) p. 30 (Nov. 1986).
20. J. Chamberlain, F. Clayton, H. Sari, and P. Vandamme, "Receiver Techniques for Microwave Digital Radio," *IEEE Communications Mag.* 24(11) p. 43 (Nov. 1986).
21. W.C. Jakes, Jr, *Microwave Mobile Communications*, Wiley-Interscience, New York (1974).
22. F. P. Duffy and T. W. Thatcher, Jr., "1969-70 Connection Survey: Analog Transmission Performance on the Switched Telecommunications Network," *BSTJ* 50(4) pp. 1311-47 (April 1971).
23. M. B. Carey, H.-T. Chen, A. Descloux, J. F. Ingle, and K. I. Park, "1982/83 End Office Connection Study: Analog Voice and Voiceband Data Transmission Performance Characterization of the Public Switched Network," *AT&T Bell Lab. Tech. J.* 63(9)(Nov. 1984).
24. D. V. Batorsky and M. E. Burke, "1980 Bell System Noise Survey of the Loop Plant," *AT&T Bell Lab. Tech. J.* 63(5) pp. 775-818 (May-June 1984).
25. B. R. Saltzberg and J.-D. Wang, "Second-order statistics of logarithmic quantization noise in QAM data communication," *IEEE Transactions on Communications* 39(10) pp. 1465-72 (Oct. 1991).
26. P. Chen, "The Compact Disk ROM: How It Works," *IEEE Spectrum* 23(4) p. 44 (April 1986).
27. S. Miyaoka, "Digital Audio is Compact and Rugged," *IEEE Spectrum* 21(3) p. 35 (March 1984).
28. P. J. Bloom, "High-Quality Digital Audio in the Entertainment Industry," *IEEE ASSP Magazine* 2(4) p. 2 (Oct. 1985).
29. J. C. Mallinson, "A Unified View of High Density Digital Recording Theory," *IEEE Trans. on Magnetics* MAG-11 p. 1166 (Sep. 1975).
30. H. Kobayashi, "A Survey of Coding Schemes for Transmission or Recording of Digital Data," *IEEE Trans. on Communications* COM-19 p. 1087 (Dec. 1971).
31. H. Bertram, "Fundamentals of the Magnetic Recording Process," *IEEE Proceedings* 74(11) p. 1494 (Nov. 1986).
32. D. Marcuse, *Light Transmission Optics*, Van Nostrand Reinhold, Princeton, N.J. (1972).
33. D. Marcuse, *Theory of Dielectric Optical Waveguides*, Academic Press, New York (1974).
34. D. Gloge, *Optical Fiber Technology*, IEEE Press, New York (1976).
35. H. H. Unger, *Planar Optical Fibers for Transmission*, Clarendon Press, Oxford (1977).
36. J. E. Midwinter, *Optical Fibers for Transmission*, Wiley, New York (1979).

37. S. E. Miller and A.G. Chynoweth, *Optical Fiber Telecommunications*, Academic Press, New York (1979).
38. H. F. Taylor, *Fiber Optics Communications*, Artech House, Dedham, Mass. (1983).
39. M. K. Barnoski, *Fundamentals of Optical Fiber Communications*, Academic Press, New York (1976).

# 6

---

## MODULATION

---

An information-bearing signal must conform to the limitations of its channel. While the bit streams we wish to transmit are inherently discrete-time, all the physical media considered in Chapter 5 are continuous-time in nature. Hence, we need to represent the bit stream as a continuous-time signal for transmission, a process called *modulation*.

This chapter describes the most common modulation and demodulation techniques, which are not necessarily optimal. Optimization often involves practical difficulties that add significantly to the cost of an implementation. For this reason, in this chapter we give a practical engineering perspective, covering only ideas that are essential in actual implementations, and deferring most issues of optimization to subsequent chapters.

We start with the basic *baseband pulse amplitude modulation (PAM)*, in which a sequence of time-translates of a basic pulse is amplitude-modulated by a sequence of data symbols. Baseband PAM is commonly used for metallic media, such as wire pairs, where the signal spectrum is allowed to extend down to zero frequency (d.c.). We then extend PAM to *passband transmission* by introducing a sinusoidal carrier signal. Passband PAM is commonly used on media with highly constrained bandwidth, such as radio. It uses two sinusoidal carriers of the same frequency (with a ninety degree phase difference) which are modulated by the real and imaginary parts of a complex-valued baseband signal. Special cases of passband PAM are the commonly used *phase-shift keying (PSK)*, *amplitude and phase modulation (AM-PM)* and

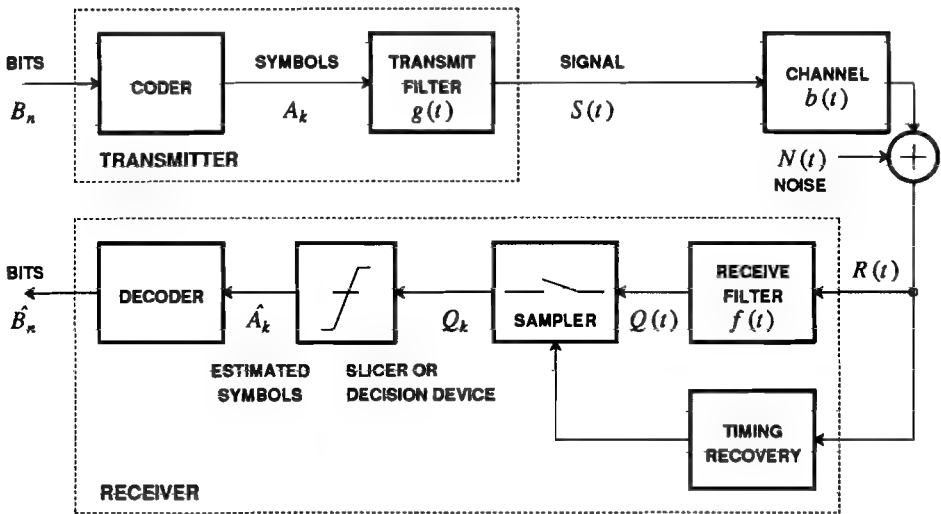
*quadrature amplitude modulation* (QAM). By treating these techniques as special cases of passband PAM we avoid the alphabet soup that pervades most comprehensive treatments of digital communications, where every minor variation is given a new acronym and treated as a separate topic. We then generalize these modulation techniques further to allow the bit stream to be mapped into a set of orthogonal waveforms, introducing a technique called *orthogonal multipulse*. A special case of this, *frequency shift keying* (FSK), is of practical importance. FSK is used when simple, inexpensive transceivers are required, when the difficulty of synchronizing the receiver to the carrier mandates that an *incoherent* receiver be used, or when the channel has significant nonlinearities. Orthogonal multipulse is then combined with PAM. Two practical examples of this combination, *multicarrier modulation* and *code-division multiplexing*, are then described. Finally, we briefly consider special features of optical fiber and magnetic recording channels in the context of the techniques described. These media are different enough to warrant special consideration.

In the past, metallic media such as wire pairs and coaxial cable have dominated the digital communications world, but this is changing rapidly. Optical fiber has rapidly assumed the role formerly played by metallic media, while digital radio provides wireless communication. For optical fiber, there is little motivation to conserve bandwidth. Thus, simple modulation techniques such as binary PAM (in the form of *on-off keying* or OOK) are commonly used. Radio channels, such as digital microwave radio, satellite, and mobile radio, are highly bandwidth constrained, and hence there is motivation to conserve bandwidth as much as possible. The voiceband telephone channel has strictly constrained bandwidth as well. When conserving bandwidth is of paramount importance, more sophisticated modulation techniques such as PSK, QAM, and multicarrier are commonly used. In this chapter we emphasize these bandwidth-conserving techniques.

## 6.1. AN OVERVIEW OF BASIC PAM TECHNIQUES

A complete baseband digital communication system is shown in Figure 6-1. In this section we describe qualitatively the structure of such a system. Every component is discussed in detail in other parts of the book. Channels have already been discussed in Chapter 5, although we will again summarize their characteristics briefly here. *Transducers* (such as lasers for optical fibers or antennas for microwave) are assumed to be part of the channel.

Typically, a transmitter and receiver will be packaged together so that communication in both directions can be performed. Such a package is called a *modem*, which stands for modulator/demodulator. Often the transmit and receive signal share the same physical medium. This is a special case of *multiple access*, described in part V of this book.



**Figure 6-1.** A baseband digital communication system, showing transmit coder, transmit filter, channel, receive filter, sampler and timing recovery, decision, and decoder.

### 6.1.1. Channel

The characteristics of channels based on common transmission media were discussed in Chapter 5. With a couple of exceptions, these channels are adequately modeled as a linear time-invariant filter with impulse response  $b(t)$  and additive noise  $N(t)$ , as shown in Figure 6-1. The major exceptions to the linear channel model apply to microwave radio channels and the magnetic recording channel. A major exception to the additive noise model is the signal shot noise encountered in optical fiber channels. These channels therefore require special techniques, to be described separately.

The media of Chapter 5 all have noise, and most can be modeled by additive noise  $N(t)$  as shown in Figure 6-1. In most cases the noise can be considered Gaussian because its origin is thermal. In many other cases, as in shot noise in optical communications systems, this noise is often approximated as Gaussian.

#### Example 6-1.

The crosstalk between metallic cable pairs is distinctly non-Gaussian interference. However, in many applications there are a large number of independent interferers, and by the central limit theorem the combined crosstalk will be approximately Gaussian.  $\square$

#### Example 6-2.

An optical signal displays considerable non-Gaussian randomness due to quantum effects. However, at practical signal levels these quantum effects can usually be approximated as Gaussian. Furthermore, we will see that the thermal noise introduced in the receiver circuitry, rather than the quantum noise, is often the most significant disturbance. Sometimes, however, the noise in an optical system cannot be modeled as Gaussian; these cases will be

discussed in Chapter 8.  $\square$

### 6.1.2. Transmitter

As shown in Figure 6-1, an incoming bit stream is fed to a *coder*, which converts the incoming bit stream into a stream of *symbols*. While a bit can only assume the values "0" or "1", a symbol assumes values from an *alphabet* that we can define.

#### Example 6-3.

The simplest coder translates the bits into symbols with the same values, so the alphabet is  $\{0,1\}$ . A slightly more complicated coder might use alphabet  $\{-1,1\}$  so that the symbols have zero mean if the bits are equally likely to be "0" and "1". A more complicated coder might map pairs of bits from the set  $\{00,01,10,11\}$  into one of four levels from the alphabet  $\{-3,-1,1,3\}$ . Another coder maps the set  $\{00,01,10,11\}$  into complex-valued symbols  $\{+1,+j,-1,-j\}$  (this applies to the passband case). All of these coders are used in practice.  $\square$

Since the coder may map multiple bits into a single data symbol, we must make a distinction between the *symbol rate* and the *bit rate*. The symbol rate is also called the *baud rate*, after the French telegraph engineer Baudot.

#### Example 6-4.

If the coder maps two bits into a symbol with an alphabet size of four, the symbol rate is half the bit rate.  $\square$

In the examples thus far, there is a one-to-one mapping between blocks of input bits and the alphabet. A coder may also increase the alphabet size, usually in order to introduce *redundancy*. For example, the coder might convert an input bit into a symbol from an alphabet of size three. Alternatively, the coder could convert an input bit into a sequence of two or more symbols, in which case the symbol rate would be higher than the bit rate. These possibilities are discussed in Chapters 12, 13 and 14, where it is shown that redundancy can be used to reduce errors or control the power spectrum. For the purposes of this chapter, we will assume that the coder does not introduce redundancy. Specifically, we will usually assume that the symbols coming from the coder are independent and identically distributed, forming a white discrete-time random process.

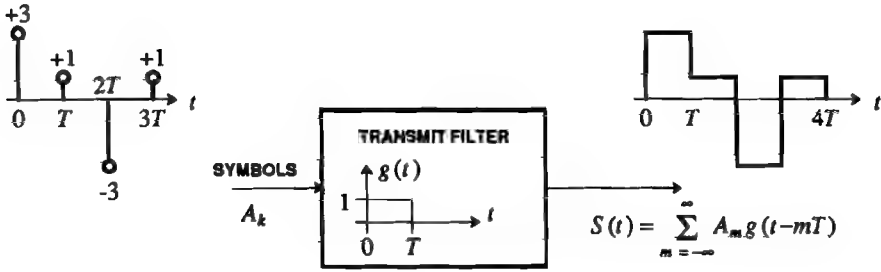
Symbols are applied to a *transmit filter*, which produces a continuous-time signal for transmission over the continuous-time channel.

#### Example 6-5.

A simple transmit filter has a rectangular impulse response, shown in Figure 6-2. The signal produced has very wide bandwidth, however, so it is not suitable for bandlimited channels.  $\square$

The impulse response  $g(t)$  of the transmit filter is called the *pulse shape*. The output of the transmitter is the convolution of the pulse shape with the symbol sequence,





**Figure 6-2.** A transmit filter with a rectangular impulse response. The symbol rate is  $1/T$  symbols per second. A sample symbol sequence (with alphabet size of four) and corresponding continuous-time signal are also shown.

$$S(t) = \sum_{m=-\infty}^{\infty} A_m g(t-mT), \quad (6.1)$$

where  $1/T$  is the symbol rate. This signal can be interpreted as a sequence of possibly overlapped pulses with the amplitude of each determined by a symbol. Such signals are termed *pulse amplitude modulated* (PAM) signals, regardless of the pulse shape. PAM and its generalization to passband are by far the most common signaling methods in digital communications. There is a confusing array of techniques (e.g., QAM, PSK, BPSK, PRK, QPSK, DPSK, and AM-PM) which are all special cases of passband PAM, perhaps with some special coding. We further generalize PAM to include FSK and multicarrier techniques in Section 6.6 and beyond.

A linear channel with impulse response  $b(t)$  and additive noise  $N(t)$  will result in a received signal

$$R(t) = \int_{-\infty}^{\infty} b(\tau) \sum_{m=-\infty}^{\infty} A_m g(t-mT-\tau) d\tau + N(t). \quad (6.2)$$

This can be rewritten as

$$R(t) = \sum_{m=-\infty}^{\infty} A_m h(t-mT) + N(t), \quad (6.3)$$

where  $h(t)$  is the convolution of  $b(t)$  with  $g(t)$ ,

$$h(t) = \int_{-\infty}^{\infty} b(\tau) g(t-\tau) d\tau \quad (6.4)$$

and is called the *received pulse*.

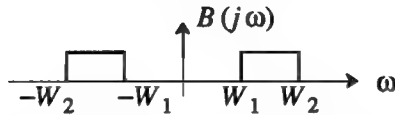
#### Example 6-6.

If the channel is ideally bandlimited, so that the transfer function is

$$B(j\omega) = \begin{cases} 1; & |\omega| < W \\ 0; & |\omega| \geq W \end{cases} \quad (6.5)$$

then the pulse shape of the previous example may not be practical because it will be severely distorted by the channel (see Problem 6-1). The design of more appropriate pulses is considered in Section 6.2.  $\square$

If the channel is an ideal bandpass channel (or some approximation), like that shown below:



then it is usually necessary to *modulate* the PAM signal. This is called *passband PAM*, and is considered in Section 6.4.

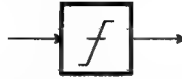
### 6.1.3. Receiver

The design of a good receiver, one that minimizes the probability of error, is important and complicated enough to dominate the design effort as well as this book. Here we introduce the basic components of the receiver.

The receiver needs to extract the discrete-time information from the continuous-time waveform which has been corrupted by the channel. To construct a discrete-time signal from a continuous-time signal, we need to know the appropriate rate at which to sample the continuous-time signal. In other words, to estimate the symbol sequence, we need to know the precise frequency and phase of the transmitted data symbols as corrupted by the channel. This is known as the *timing information*. For useful communication systems the transmitted signal originates at some other geographic location, or at some other time, so the timing of the transmitter is not directly available. Either a timing signal must accompany the data-bearing signal, or much more commonly the timing must be extracted from the data-bearing signal. In either case, the *timing recovery* component in Figure 6-1 produces a synchronization signal which is used to convert the continuous-time signal into a discrete-time signal. Timing recovery methods are discussed in Chapter 17. Until that chapter, we will simply ignore the issue and assume that the receiver is somehow synchronized to the transmitter.

The first processing done by the receiver is usually an *automatic gain control*, or AGC. This is not shown because we assume its effect is transparent. The receiver next filters the incoming signal  $R(t)$ . The *receive filter* can perform several functions, such as compensating for the distortion of the channel and diminishing the effect of additive noise. The optimal design of such filters is considered in detail in subsequent chapters. In Chapter 11, adaptive techniques are described, in which the receive filter transfer function is adaptively adjusted to learn an unknown channel or to track changing channel responses. Adaptive filtering allows a modem to be designed without knowledge of the specific channel over which it will operate, since an adaptive receiver can learn the channel characteristics after it is deployed in the field.

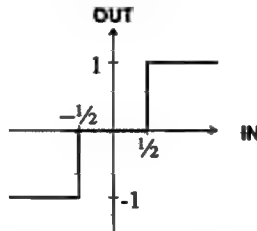
The second stage in the receiver is a sampler, which converts the signal to a discrete-time signal at the sample rate equal to the symbol rate with a frequency and phase determined by the timing recovery. Ideally, the continuous-time signal is sampled at a point where the sample equals the transmitted symbol, with minimum interference from neighboring symbols. The third stage is the decision, in which an estimate  $\hat{A}_k$  of the symbol sequence is constructed. The decision device is usually a quantizer (or *slicer*), shown symbolically below:



The slicer applies a series of *decision thresholds* to the input signal.

**Example 6-7.**

If the data symbols are drawn from the alphabet  $\{-1, 0, 1\}$ , then the slicer would typically apply decision thresholds at  $-\frac{1}{2}$  and  $\frac{1}{2}$  as shown below:



The output of the slicer is a *decision* as to the transmitted data symbol in that symbol interval.  $\square$

The design of the slicer is described in Section 6.5 and in subsequent chapters. Throughout this chapter we assume a common-sense slicer which is usually so close to the optimum in performance that there is little to be gained from using the optimum. Optimal design is deferred to subsequent chapters.

Finally, the estimated symbol sequence is decoded to produce a bit stream. This decoder performs the inverse mapping to the coder in the transmitter.

In the sequel we will see many variations of this basic receiver structure. For example, the receive filter may be partially implemented in discrete time, or the decision and decoding operations may be performed together.

### 6.1.4. Overview of Performance Measures

A basic performance measure of a digital communication system is the bit rate achieved for a given channel, where the objective is usually to maximize that bit rate. In Chapter 4, we derived the capacity of discrete-time channels with additive white Gaussian noise, finding that each symbol (one "channel use") can reliably convey an amount of information given by

$$C_s = \frac{1}{2} \log_2 \left( 1 + \frac{\sigma_x^2}{\sigma^2} \right) \quad (6.6)$$

where  $\sigma^2$  is the variance of the additive Gaussian noise samples and  $\sigma_x^2$  is the variance of the received symbols. This capacity is measured in bits per symbol. If the channel can carry complex-valued symbols, then the capacity per symbol is twice that in (6.6). We will often model passband channels as complex baseband equivalent channels.

In this chapter, we will determine the rate at which symbols can be transmitted without interfering with one another. While (6.6) promises error-free transmission at this rate, it can be achieved only with arbitrarily large transmitter and receiver complexity and unbounded processing delay. This chapter will develop practical and simple transmitters and receivers, for which we have to accept some non-zero probability of incorrectly detecting the symbols. This probability of error must be kept acceptably low while maximizing the bit rate. Often, the complexity (cost) of the receiver can be traded off against the bit rate for a given channel if the probability of error is held constant.

#### Example 6-8.

For the telephone channel, an acceptable probability of error is usually about  $10^{-6}$ . A very inexpensive modem that easily meets this requirement uses frequency shift keying (FSK) (described in Section 6.6) to achieve a rate of 300 bits per second (b/s). Phase-shift keyed (PSK) (Section 6.4 and 6.5) modems achieve 1200 b/s. Quadrature amplitude modulation (QAM) (Section 6.4 and 6.5) is used to get 2400 b/s. QAM modems are ubiquitous and inexpensive, despite the use of relatively sophisticated techniques such as adaptive equalization (Chapter 11). The full-duplex modems mentioned thus far divide the telephone bandwidth (about 3000 Hz) into two bands (about 1 kHz each), one band for each direction of transmission. Higher speeds, such as 9600 b/s are achieved by using all or most of the telephone bandwidth for each direction of transmission and either separating the two signals using echo cancellation (Chapter 19) or alternating transmission in each direction. Coded modulation (Chapter 14) is used to get closer to the theoretical channel capacity. This adds a great deal to the complexity of a modem, although the speed required of the hardware is still low enough for practical low-cost implementation. The channel capacity of a telephone channel has been estimated to be over 30,000 b/s (Section 5.5 and Chapter 8), with the most elaborate modems getting impressively close to this.  $\square$

For a practical communication system, such as that in Figure 6-1, the bit rate is determined by the symbol rate, the size of the alphabet, and the coder. If we denote the alphabet by  $\Omega_A$  and its size by  $|\Omega_A|$ , and if the coder maps blocks of bits one-to-one into symbols, then  $\log_2 |\Omega_A|$  bits are transmitted per symbol.

#### Example 6-9.

If the alphabet is  $\{-3, -1, +1, +3\}$  then we can transmit two bits per symbol.  $\square$

In general, a higher symbol rate requires more channel bandwidth. The bit rate achieved in a given channel bandwidth can be quantified using *spectral efficiency* [1], defined as

$$\nu = \frac{\text{bit rate}}{\text{bandwidth in Hz}} . \quad (6.7)$$

Spectral efficiency has the units of bits/sec-Hz.

**Example 6-10.**

In point-to-point microwave radio, the spectrum is a public resource and it is important to use it efficiently for maximum public benefit. Therefore, the regulatory agencies have placed minimum requirements on the bit rates achieved as well as the bandwidth allowed. For example, 500 MHz of bandwidth is allocated in the United States for digital radio at 4 GHz divided into channels with a spacing of 20 MHz. Each channel is required to carry 90 Mb/s, for a spectral efficiency of 4.5 bits/sec-Hz [2]. Higher efficiencies would of course be desirable.  $\square$

**Example 6-11.**

Voiceband data modems are available at bit rates as high as 28.8 kb/s. If the bandwidth is 3.2 kHz, then the spectral efficiency is 9 bits/sec-Hz. Lest the reader conclude that the voiceband data designers must be smarter, it should also be noted that the digital radio operates at bit rates about 5000 times faster, making the implementation problems somewhat more severe. Also, the SNRs for the two systems are different.  $\square$

For an alphabet of size  $|\Omega_A|$ , if the coder maps blocks of bits one-to-one into symbols, then  $\log_2 |\Omega_A|$  bits are transmitted per symbol, so the spectral efficiency will be

$$\nu = \frac{\log_2 |\Omega_A|}{BT} , \quad (6.8)$$

where  $B$  is the bandwidth in Hz and  $T$  is the symbol interval ( $1/T$  is the symbol rate).

**Example 6-12.**

Low-speed voiceband data modems conforming to the V.22bis CCITT standard transmit 2400 b/s in each direction using a 16 symbol alphabet at a symbol rate of  $1/T = 600$  symbols/sec. They use half the channel for each direction of transmission, using a signal bandwidth of about  $B = 1200$  Hz. Hence

$$\nu = \frac{\log_2 |\Omega_A|}{2} = 2 . \quad (6.9)$$

To achieve a spectral efficiency of 4 bits/sec-Hz at the same symbol rate and bandwidth would require

$$\log_2 |\Omega_A| = 8 \quad (6.10)$$

implying a large alphabet with 256 symbols.  $\square$

The symbol rate is bounded by the bandwidth constraints of the channel. If we wish to avoid interference between symbols, then symbols may be transmitted at a rate no greater than twice the bandwidth of the channel, as shown in Section 6.2 below. Furthermore, the size of the alphabet is constrained by the allowable transmitted power and by the additive noise on the channel, although this constraint can often be relaxed (to a limit) by more complex receiver processing. The combination of these two constraints — on symbol rate and alphabet size — limits the available bit rate for

a given channel.

## 6.2. PULSE SHAPES

Suppose that data symbols  $A_k$  have power (variance)  $E |A_k|^2 = \sigma_A^2$ . Further, assume successive data symbols are uncorrelated. Then the discrete-time random process  $\{A_k, -\infty < k < \infty\}$  is white, and has power spectrum

$$S_A(e^{j\omega T}) = \sigma_A^2. \quad (6.11)$$

(Recall from Chapter 3 that we use the notation  $S_A(e^{j\omega T})$  to indicate the power spectrum of a discrete-time signal, and the notation  $S_S(j\omega)$  for the power spectrum of a continuous-time signal  $S(t)$ .) Consider the baseband PAM signal

$$S(t) = \sum_{m=-\infty}^{\infty} A_m g(t + \Theta - mT) \quad (6.12)$$

where  $\Theta$  is an unknown phase. Through most of this chapter we will assume that  $\Theta = 0$ . However, for purposes of determining the power spectrum of the transmitted signal, in order for  $S(t)$  to be wide-sense stationary we must assume that  $\Theta$  is uniformly distributed over the interval  $[0, T)$ . The unknown phase simply reflects the fact that the origin of the time axis is arbitrary. From Appendix 3-A, the power spectrum of  $S(t)$  is

$$S_S(j\omega) = \frac{1}{T} |G(j\omega)|^2 S_A(e^{j\omega T}) = \frac{\sigma_A^2}{T} |G(j\omega)|^2. \quad (6.13)$$

Thus the shape of the power spectrum of the transmitted signal is determined by the transfer function  $G(j\omega)$  of the transmit filter. In practice we design the pulse shape  $g(t)$  to meet power spectrum constraints on the channel. The channel may be bandlimited, for example, or may lack a d.c. component. This power spectrum also affects the interference into other channels, such as *radio-frequency interference (RFI)* or *crosstalk*.

If the channel is an ideal bandlimited channel,

$$B(j\omega) = \begin{cases} 1; & |\omega| < W \\ 0; & |\omega| \geq W \end{cases} \quad (6.14)$$

then the ideally bandlimited pulse can be used,

$$G(j\omega) = \begin{cases} \pi/W; & |\omega| < W \\ 0; & |\omega| \geq W \end{cases} \quad (6.15)$$

which in the time domain is a sinc pulse

$$g(t) = \frac{\sin(Wt)}{Wt} \quad (6.16)$$

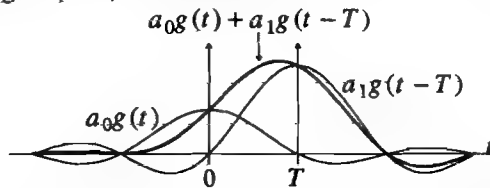
as shown below:



Notice that the pulse has zero crossings at all multiples of  $\pi/W$  except at  $t = 0$ , where  $g(0) = 1$ . This implies that if we set  $T = \pi/W$ , and sample the signal (6.12) at times  $t = mT$  for integers  $m$ , the result is the symbol sequence  $A_m$  (assuming phase  $\Theta = 0$ ).

#### Example 6-13.

Consider two successive symbols with values  $a_0 = 1$  and  $a_1 = 2$ . The contribution of these two symbols to the signal (6.12) is shown below:



If the channel is given by (6.14), then the receiver only needs to sample at 0 and  $T$ . Neighboring symbols do not interfere with one another at the proper sampling time, so we say that there is no *intersymbol interference (ISI)*.  $\square$

We will see in Section 6.2.1 that this choice of pulse shape maximizes the rate at which symbols can be transmitted over a bandlimited channel without ISI. However, it is not a practical pulse shape, since ideally bandlimited transfer functions are not realizable. Even close approximations to this ideally bandlimited pulse are undesirable, as shown in Chapter 17, because timing recovery becomes very difficult. We consider below more practical pulse shapes which use more than the minimum bandwidth, and in the process establish the minimum bandwidth required for pulse transmission.

### 6.2.1. Nyquist Pulses

We saw above that, in principle, an ideally bandlimited pulse can be used to transmit symbols, and the symbols can be recovered by sampling the signal. From (6.16) and Example 6-13 we see that the ideal bandlimited pulse has bandwidth  $W = \pi/T$ , where  $T$  is the symbol period ( $1/T$  is the symbol rate). In other words,  $G(j\omega) = 0$  for  $\omega > \pi/T$ , so from (6.13) the signal  $S(t)$  is bandlimited to  $W = \pi/T$ . We will see shortly that this is the minimum bandwidth for a fixed  $T$  so that the signal can be sampled to recover the symbols.

Minimum bandwidth is desirable, but the ideal bandlimited pulse is impractical. Therefore, practical systems use pulses with more bandwidth than the ideal bandlimited pulse. The bandwidth above the minimum is called *excess bandwidth*. Usually excess bandwidth is expressed as a percentage; for example, 100% excess bandwidth corresponds to a bandwidth of  $2\pi/T$ , or twice the minimum. Practical systems usually have an excess bandwidth in the range of 10% to 100%. Increasing the excess bandwidth simplifies implementation (simpler filtering and timing recovery), but of

course requires more channel bandwidth.

The zero-excess-bandwidth pulse is unique — the ideal bandlimited pulse of the last section. With non-zero excess bandwidth, the pulse shape is no longer unique. In this subsection we derive a criterion, called the *Nyquist criterion*, that must be met by received pulses if there is to be no intersymbol interference, and illustrate some pulse shapes that satisfy this criterion, called *Nyquist pulses*.

The input to the sampler can be written as

$$Q(t) = \sum_{m=-\infty}^{\infty} A_m p(t-mT) + U(t) \quad (6.17)$$

where the filtered noise process is

$$U(t) = N(t) * f(t) \quad (6.18)$$

and the pulse shape at the slicer is

$$p(t) = g(t) * b(t) * f(t), \quad (6.19)$$

where  $b(t)$  is the impulse response of the channel and  $f(t)$  is the impulse response of the receive filter. We have assumed the random phase  $\Theta$  is zero. Sampling (6.17) yields

$$\begin{aligned} Q_k &= \sum_{m=-\infty}^{\infty} A_m p(kT - mT) + U(kT) \\ &= A_k p(0) + \sum_{m \neq k} A_m p(kT - mT) + U(kT). \end{aligned} \quad (6.20)$$

The second term is called the *intersymbol interference* (ISI). If  $p(t)$  crosses zero at non-zero multiples of  $T$ ,

$$p(kT) = \delta_k \quad (6.21)$$

for all integers  $k$ , then the output of the sampler is

$$Q_k = Q(kT) = \sum_{m=-\infty}^{\infty} A_m \delta_{k-m} + U_k = A_k + U_k \quad (6.22)$$

where  $U_k = U(kT)$ . In this case there is no ISI. We rely on the timing recovery (described in Chapter 17) to supply the correct sampling instants,  $t = kT$ .

Usually we want to design pulses that avoid ISI. (An exception is partial response signaling, described in Chapter 12, in which ISI is deliberately introduced.) How can we do this without using the ideal bandlimited pulse? For a given impulse response  $b(t)$  of the channel, we can design  $g(t)$  and  $f(t)$  to force correct zero crossings in  $p(t)$ . This criterion on  $p(t)$  in (6.21) is called the *zero-forcing (ZF)* criterion, because it forces the ISI to zero. It is not necessarily optimal because it ignores the effect of the noise; forcing the ISI to zero may increase the noise. Joint minimization of ISI and noise is explored in Chapter 10.

For low noise levels, we clearly wish to approximate the ZF criterion. To get zero ISI it is necessary that (6.21) be satisfied. Taking the Fourier transform of each



side of (6.21) and using (2.17) we see that

$$\frac{1}{T} \sum_{m=-\infty}^{\infty} P(j\omega - jm\frac{2\pi}{T}) = 1. \quad (6.23)$$

This is called the *Nyquist criterion*. The minimum-bandwidth pulse satisfying (6.23) is the ideal lowpass pulse, the sinc function, so we have demonstrated that a bandwidth of at least  $W = \pi/T$  is required for zero ISI. Put another way, if we are constrained to frequencies  $|\omega| < W$ , the maximum symbol rate  $1/T$  that can be achieved with zero ISI is  $1/T = W/\pi$ .

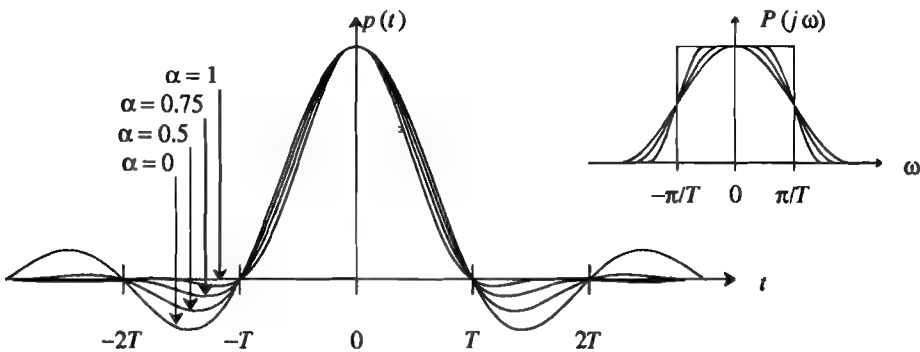
Commonly used pulses  $p(t)$  that satisfy the Nyquist criterion are the *raised-cosine pulses*, given by

$$p(t) = \left[ \frac{\sin(\pi t/T)}{\pi t/T} \right] \left[ \frac{\cos(\alpha \pi t/T)}{1 - (2\alpha t/T)^2} \right] \quad (6.24)$$

which have Fourier transforms

$$P(j\omega) = \begin{cases} T; & 0 \leq |\omega| \leq (1-\alpha)\pi/T \\ \frac{T}{2} \left[ 1 - \sin \left[ \frac{T}{2\alpha} (|\omega| - \frac{\pi}{T}) \right] \right]; & (1-\alpha)\frac{\pi}{T} \leq |\omega| \leq (1+\alpha)\frac{\pi}{T} \\ 0; & |\omega| > (1+\alpha)\pi/T \end{cases} \quad (6.25)$$

These pulses and their Fourier transforms are plotted in Figure 6-3 for a few values of  $\alpha$ . For  $\alpha = 0$ , the pulse is identical to the ideally bandlimited pulse (6.16). For other values of  $\alpha$ , the energy rolls off more gradually with increasing frequency, so  $\alpha$  is called the *roll-off factor*. The shape of the roll-off is that of a cosine raised above the abscissa, which explains the name. The pulse for  $\alpha = 0$  is the pulse with the smallest



**Figure 6-3.** A family of pulses with zero crossings at multiples of  $T$ , for four values of  $\alpha$ , the roll-off factor. The Fourier transform of the pulses is also shown. Note the raised-cosine shape, and the excess bandwidth that increases with  $\alpha$  from 0% to 100%.

bandwidth that has zero crossings at multiples of  $\pi/W$ ; larger values of  $\alpha$  require excess bandwidth varying from 0% to 100% as  $\alpha$  varies from 0 to 1. In the time domain, the tails of the pulses are infinite in extent. However, as  $\alpha$  increases, the size of the tails diminishes. For this reason, these pulses can be practically approximated using FIR filters by truncating the pulse at some multiple of  $T$ .

There are an infinite number of pulses that satisfy the Nyquist criterion and hence have zero crossings at multiples of  $\pi/W$ . Some examples are shown in Figure 6-4.

#### Example 6-14.

Consider a channel bandlimited to  $|\omega/2\pi| \leq 1500\text{Hz}$ . The absolute maximum symbol rate using signaling of the form (6.12) is 3000 symbols per second. If we use a pulse with 100% excess bandwidth, then the maximum symbol rate is 1500 symbols per second.  $\square$

## 6.3. BASEBAND PAM

In Figure 6-1, we are free to design  $g(t)$  and  $f(t)$ , but not  $b(t)$ . The impulse responses  $g(t)$  and  $f(t)$  can be chosen to force the ISI to zero, satisfying the zero-forcing criterion. One difficulty with exactly satisfying the ZF criterion is that the channel is rarely completely known at the time the filters are designed. Furthermore, even when the channel is known, the filters required to exactly satisfy the ZF criterion may be difficult or expensive to realize. In this section we describe practical engineering techniques for the design of baseband PAM systems.

### 6.3.1. ISI and Eye Diagrams

With suboptimal filtering, it is useful to quantify the degradation of the signal. A useful graphical illustration of the degradation is the *eye diagram*, so called because its shape is similar to that of the human eye. An eye diagram is easily generated using an oscilloscope to observe the output of the receive filter, where the symbol timing serves as the trigger. Such displays have historically served as a quick check of the performance of a modem in the field. The eye diagram is also a useful design tool during the analytical and simulation design phase of the system.

An eye diagram consists of many overlaid traces of small sections of a signal, as shown in Figure 6-5. If the data symbols are random and independent, it summarizes

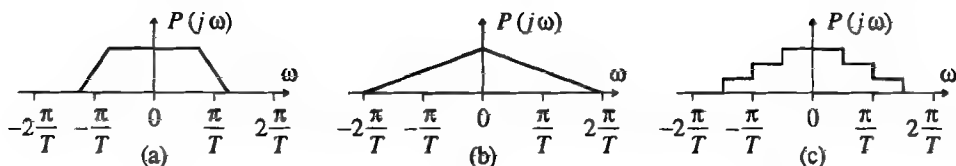
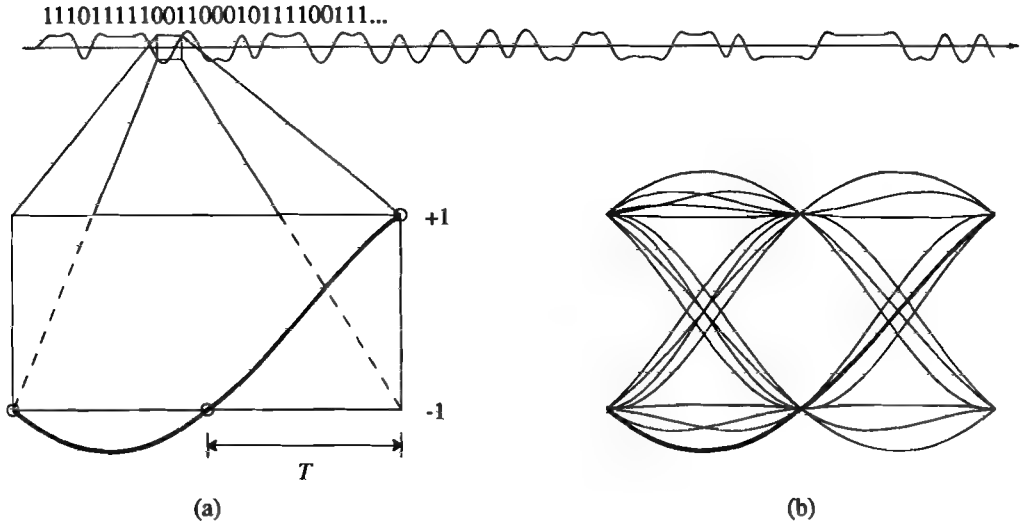
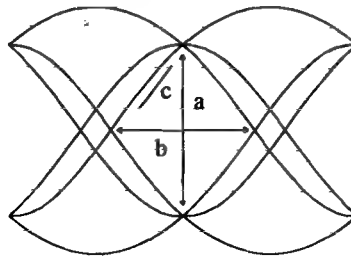


Figure 6-4. The Fourier transform of some pulses that satisfy the Nyquist criterion.



**Figure 6-5.** A binary PAM signal made with 50% excess-bandwidth raised-cosine pulses. A segment of length  $2T$  is shown in detail in (a). The small circles indicate the sample points where symbols are unperturbed by neighboring symbols. In (b), an eye diagram is made by overlaying sections of length  $2T$ . The component from part (a) is shown darkened. This display is typical of an oscilloscope display of a signal, where the oscilloscope is triggered at the symbol rate.

visually all possible intersymbol interference waveforms. It summarizes several features of the signal, as shown in Figure 6-6. In the presence of intersymbol interference, when the pulse shape does not satisfy the Nyquist criterion, the eye diagram will tend to close vertically. For error-free transmission in the absence of noise, the eye must maintain some vertical opening, since otherwise there are intersymbol interference waveforms that will cause errors.

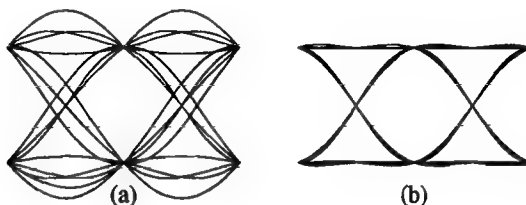


**Figure 6-6.** A summary of the salient features of an eye diagram. The vertical eye opening (a) indicates the immunity to noise. The horizontal eye opening (b) indicates the immunity to errors in the timing phase. The slope of the inside eye lid (c) indicates the sensitivity to jitter in the timing phase. The ZF criterion is satisfied if all traces pass through the two symbol values in the center of the eye.

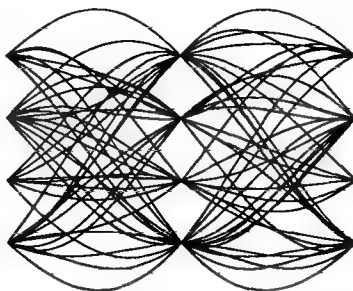
When there is incomplete vertical closure, the intersymbol interference will reduce the size of the additive noise required to cause errors. Hence, the wider the vertical opening, the greater the noise immunity. The ideal sampling instant is at the point of maximum (vertical) eye opening, but this can never be achieved precisely by a practical timing recovery circuit. Thus the horizontal eye opening is also practically important, since the smaller this opening the greater the sensitivity to errors in timing phase (the instant at which the signal is sampled).

The shape of the eye is determined by the pulse shape. In particular, the vertical eye opening is determined by the size of the pulse at multiples of  $T$ , and the horizontal eye opening is determined by the size of the tails of the pulse  $p(t)$ . In Figure 6-7 are shown two eye diagrams for 25% and 100% excess-bandwidth raised-cosine pulses. It is important to note the beneficial effect of increasing the excess bandwidth in terms of horizontal eye opening. However, more bandwidth might allow more noise to reach the decision slicer, if we do not carefully design the receive filter. Thus, there is a basic system tradeoff between excess bandwidth, noise immunity, and the complexity of the timing recovery circuitry. In particular, without special coding (see Chapter 12) it is futile to try to achieve zero excess bandwidth because the horizontal eye opening is zero (see Problem 6-5).

An eye diagram for a four-level PAM signal is shown in Figure 6-8.



**Figure 6-7.** Eye diagrams for (a) 25% and (b) 100% excess bandwidth raised-cosine pulses. Note that the pulse with larger tails and less bandwidth (25%) has a smaller eye opening.



**Figure 6-8.** An eye diagram for a baseband PAM signal made with 25% excess-bandwidth raised-cosine pulses and an alphabet with four equally spaced symbols.

### 6.3.2. Simple, Inexpensive PAM Transmitters

When the bit rate required is much smaller than the channel capacity, practical communication systems can be designed with only rudimentary attention to the ZF criterion.

#### Example 6-15.

Consider a binary transmission system with an alphabet of only two symbols. The system transmits one bit per symbol. If the incoming bits are equally likely to be one or zero, then a practical choice of alphabet is the set  $\{-a, a\}$ , where  $a$  is selected to satisfy channel power constraints. This minimizes the power of the transmitted signal for a given spacing  $d = 2a$  between symbols. A simple transmitter can be designed with digital logic and a low pass transmit filter as shown in Figure 6-9. The digital logic acts like a rectangular pulse shaping filter, which should be considered in combination with the transmit filter. The lowpass filter in Figure 6-9 is a simple RC filter with a small RC time constant. The pulse shape, eye diagram, and sample waveform are shown in Figure 6-10.  $\square$

Transmitter design can get much more elaborate. But proper design of the transmit filter depends on the receiver, so we turn our attention to the design of the receiver.

### 6.3.3. Baseband PAM Receivers

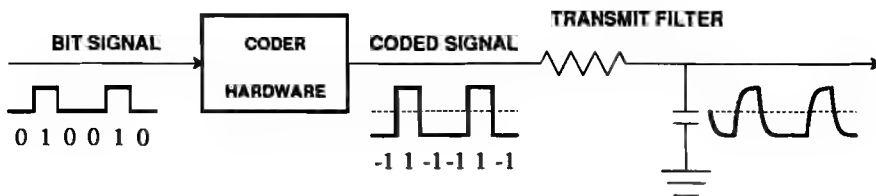
Assume a well-designed transmitter and a benign channel so that the eye is open at the receiver. A simple receiver can sample the incoming signal  $R(t)$  at the symbol rate, compare the samples against a set of thresholds, and decide which symbol is closest. Such a receiver is usually far from optimal, however, because it omits the receive filter  $f(t)$  shown in Figure 6-1. The receive filter should be used to reject components of the channel noise outside the signal bandwidth.

#### Example 6-16.

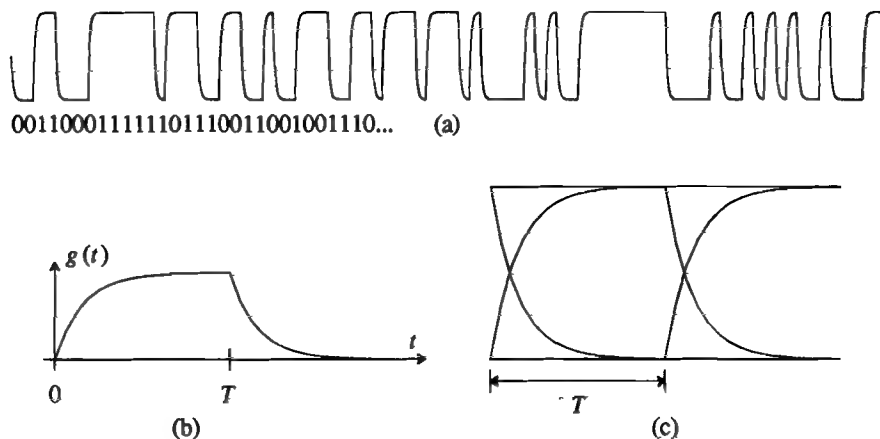
Suppose that in Figure 6-1  $N(t)$  is white Gaussian noise with power spectrum  $S_n(j\omega) = N_0$ . Suppose further that we use no receive filter, i.e.  $F(j\omega) = 1$ . If there is no ISI in the received signal, then

$$Q_k = A_k + N_k \quad (6.26)$$

where  $N_k = N(kT)$ . But from (3.186) the power  $E[|N_k|^2]$  in  $N_k$  is equal to the power in  $N(t)$ , which is infinite! The signal  $A_k$  is totally swamped by the noise!  $\square$



**Figure 6-9.** A crude baseband binary PAM transmitter that can be used when the channel capacity is much greater than the desired bit rate.



**Figure 6-10.** The first order RC filter in Figure 6-9 produces the signal shown in (a), representing the digital data shown immediately below the signal. In (b) the pulse shape is displayed for two symbol intervals. In (c) the eye diagram is shown. Note that the eye is wide open.

#### Example 6-17.

In a more realistic example,  $N(t)$  is wideband Gaussian noise with power spectrum

$$S_N(j\omega) = N_0 \text{rect}(\omega, W_N) = \begin{cases} N_0; & |\omega| \leq W_N \\ 0; & |\omega| > W_N \end{cases}, \quad (6.27)$$

where  $W_N$  is large. Its power is calculated from (3.59),

$$R_N(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_N(j\omega) d\omega = \frac{N_0 W_N}{\pi}. \quad (6.28)$$

The variance of the noise samples therefore is

$$\sigma^2 = E[|N_k|^2] = \frac{N_0 W_N}{\pi} \quad (6.29)$$

which can be quite large, depending on  $W_N$ .  $\square$

Suppose that at the receiver we can use a lowpass filter  $F(j\omega)$  without closing the eye. Its bandwidth should be as small as possible to reduce the variance of the noise samples at the slicer.

#### Example 6-18.

Assume the channel noise is that given in the previous example, but the receive filter is an ideal lowpass filter,

$$F(j\omega) = K \text{rect}(\omega, W), \quad (6.30)$$

where  $W < W_N$  and  $K$  is a normalizing constant. Define the noise component after the receive filter as

$$U(t) = N(t) * f(t). \quad (6.31)$$

Its power spectrum, from (3.64), is

$$S_U(j\omega) = S_N(j\omega) |F(j\omega)|^2 = N_0 K^2 \text{rect}(\omega, W) \quad (6.32)$$

and its power is  $WN_0K^2/\pi$ . Hence, with an ideal lowpass receive filter, the variance of the noise samples at the slicer is

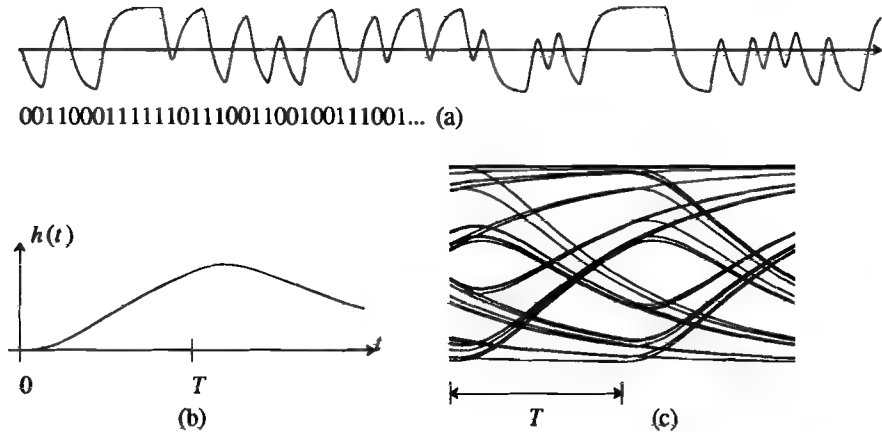
$$\sigma^2 = \frac{WN_0K^2}{\pi}. \quad (6.33)$$

These results suggest that the bandwidth  $W$  of the receive filter should be small to reduce the noise power at the slicer, but how small can we make it? If it is too small or otherwise badly designed it will affect the signal, introducing ISI.  $\square$

### Example 6-19.

Consider putting the signal in Figure 6-10 through a second-order lowpass filter with cutoff (3 dB) frequency at the symbol rate  $2\pi/T$ . The pulse shape after the filter, an example of a signal, and an eye diagram are shown in Figure 6-11. Notice that the eye is almost closed. The noise and timing phase immunity of this signal is poor.  $\square$

The task of the receive filter is to condition the signal for sampling. To avoid ISI, the resulting pulse shape  $p(t)$  should satisfy the Nyquist criterion, but at the same time the noise power admitted by the receive filter should be minimized. We are free to design the transmit filter  $G(j\omega)$ , subject to the power (or peak) constraints of the channel, and the receive filter  $F(j\omega)$ . Optimal design of these filters is deferred to subsequent chapters; here we concentrate on achieving a reasonable pulse shape  $p(t)$ .



**Figure 6-11.** The signal of Figure 6-10 has been put through a second order Butterworth filter with cutoff (3 dB) frequency at the symbol rate. The time function is shown in (a), the pulse shape at the receiver in (b), and the eye diagram in (c). Notice that the eye is relatively closed.

The pulse  $p(t)$  at the output of the receive filter has the Fourier transform

$$P(j\omega) = F(j\omega)B(j\omega)G(j\omega). \quad (6.34)$$

The received filter therefore is given by

$$F(j\omega) = \begin{cases} \frac{P(j\omega)}{B(j\omega)G(j\omega)}; & \text{for all } \omega \text{ such that } B(j\omega)G(j\omega) \neq 0 \\ 0; & \text{for all } \omega \text{ such that } B(j\omega)G(j\omega) = 0 \end{cases} \quad (6.35)$$

The receive filter frequency response can be safely set to zero for any  $\omega$  such that  $B(j\omega)G(j\omega) = 0$  because there is no signal at that frequency, so no information is lost. In practice the transmit filter  $G(j\omega)$  is often dictated by cost considerations, so our choice of  $P(j\omega)$  determines the receive filter according to (6.35).

The choice of  $P(j\omega)$  affects the performance of the system because of its effect on the noise. In Chapter 8 we determine the impact of the noise on the probability of error. Not surprisingly, the probability of error decreases monotonically as the *signal to noise ratio* (SNR) increases. The SNR at the slicer is the power of the signal component in  $Q_k$  divided by the power of the noise component in  $Q_k$ . We write the output of the receive filter as

$$Q(t) = \sum_{m=-\infty}^{\infty} A_m p(t - mT) + U(t) \quad (6.36)$$

where  $U(t) = N(t) * f(t)$ , so

$$Q_k = \sum_{m=-\infty}^{\infty} A_m p(kT - mT) + U_k \quad (6.37)$$

where  $U_k = U(kT)$ . If  $p(t)$  satisfies the Nyquist criterion,  $p(kT - mT) = \delta_{k-m}$  and

$$Q_k = A_k + U_k. \quad (6.38)$$

The SNR is therefore

$$SNR = \frac{E[|A_k|^2]}{\sigma^2} \quad (6.39)$$

where  $\sigma^2$  is the power of  $U_k$  (the variance of its samples). We often assume that the symbols are normalized so that  $E[|A_k|^2] = 1$ , in which case

$$SNR = \frac{1}{\sigma^2}. \quad (6.40)$$

#### Exercise 6-1.

Show that



$$\sigma^2 = \frac{1}{2\pi} \int_{\Gamma} S_N(j\omega) \left| \frac{P(j\omega)}{B(j\omega)G(j\omega)} \right|^2 d\omega, \quad (6.41)$$

where  $\Gamma$  is the region over which  $B(j\omega)G(j\omega) \neq 0$ .  $\square$

Notice from (6.35) that part of the function of the receive filter is to compensate for channel distortion  $B(j\omega)$  within the frequency band of interest. A receive filter is often called an *equalizer* because it compensates for (equalizes) the channel response. While the receive filter can eliminate ISI, there is a price to be paid in *noise enhancement*. In frequency regions where  $B(j\omega)G(j\omega)$  is small but not zero, and  $P(j\omega)$  is not small, the filter will have a large gain, which will amplify the noise and increase the probability of error. This can result from a poor choice of  $P(j\omega)$  for a given channel, but sometimes it is unavoidable if  $P(j\omega)$  is to satisfy the Nyquist criterion. In this latter case, we can view this noise enhancement as a penalty paid for having a channel that introduces ISI. In Chapter 10 we show that a *decision feedback equalizer* or *Viterbi detector* can reduce or in some cases eliminate this noise enhancement entirely. These receivers are nonlinear, and have a fundamentally different structure from that in Figure 6-1.

Another problem that often arises is a channel response which is not precisely known, or which is time-varying. This problem can be handled with an *adaptive equalizer*, as discussed in Chapter 11.

### 6.3.4. Discrete-Time Equivalent Channel

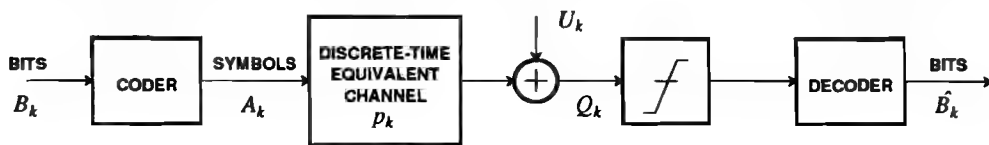
From Figure 6-1 the transmit filter, channel, and receive filter can be modeled as a single continuous-time filter. Further, since the input to this filter is discrete-time, and the output is sampled, we can replace this filter plus the sampler with an equivalent discrete-time filter, as illustrated in Figure 6-12. In the figure,

$$p_k = p(kT) = [g(t) * b(t) * f(t)]_{t=kT} \quad (6.42)$$

and

$$U_k = U(kT) = [N(t) * f(t)]_{t=kT}. \quad (6.43)$$

The receiver may consist simply of a slicer and decoder, as shown, if the eye before



**Figure 6-12.** A baseband PAM system can be modeled as an entirely discrete-time system if all continuous-time subsystems are considered to be part of the discrete-time equivalent channel.

sampling is acceptably open. Alternatively, more complicated adaptive filters (Chapter 11) are usually implemented as discrete-time filters placed after sampling and before the slicer. The discrete-time channel model summarizes all we need to know about the continuous-time portions of the system (transmitter, channel, and receiver) for purposes of computing the probability of error (see Chapter 8) and designing adaptive equalizers.

## 6.4. PASSBAND PAM

Many practical communication channels do not support transmission of baseband signals. Most physical transmission media are incapable of transmitting frequencies at d.c. and near d.c., whereas baseband PAM signals as discussed in the last section usually contain d.c. and low-frequency components.<sup>1</sup>

### Example 6-20.

Telephone channels, designed for voice, carry signals in the frequency range of about 300–3300 Hz with relatively little distortion. Radio channels are restricted to specified frequency bands by government regulatory bodies, such as the Federal Communications Commission (FCC) in the U.S., and constrain these channels to a bandwidth which is small relative to the center frequency.  $\square$

The development in this section could proceed in two ways: we could consider the transmitted signal to be a random process or a deterministic signal. Since the deterministic model is more intuitive, this will be our approach. In other words, we assume that the transmitted symbol sequence is known. The random model leads to similar results, and is analyzed in appendix 6-A.

### 6.4.1. Modulation Techniques

Assume that the sequence  $a_k$  of transmitted symbols that we wish to transmit is known. It can be considered to be an outcome of the random process  $A_k$  used above. Consistent with our notation in Chapter 3, we denote the outcome by  $s(t)$  instead of the random process  $S(t)$ . For mathematical reasons, we also need to assume that the symbol sequence is finite,

$$a_k = 0; \text{ for } |k| > M. \quad (6.44)$$

This ensures the existence of the Fourier transforms that we will need.  $M$  can be arbitrarily large, so this assumption is not seriously restrictive.

Assume that the real-valued passband signal to be transmitted over the channel is  $x(t)$ . It was shown in Section 2.4 that any such passband signal can be represented in terms of an equivalent complex-valued baseband signal  $s(t)$ , where

---

<sup>1</sup> As shown in Chapter 12, the d.c. component can be removed by line coding, but components near d.c. usually remain. These cannot be tolerated by many channels.

$$x(t) = \sqrt{2} \cdot \text{Re} \{ s(t) e^{j\omega_c t} \} . \quad (6.45)$$

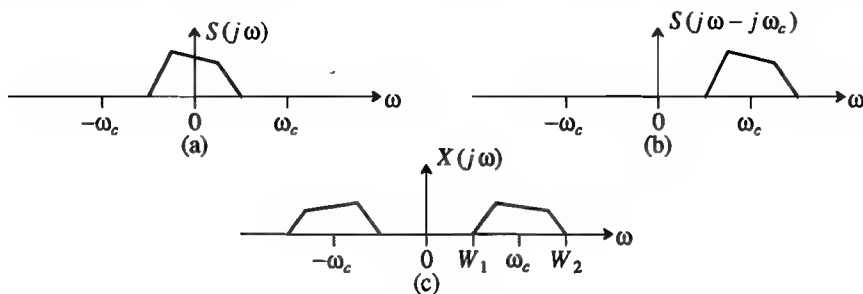
The relationship between these signals is illustrated in the frequency domain in Figure 6-13. The  $\sqrt{2}$  factor is included to force the energy of  $s(t)$  and  $x(t)$  to be the same. The frequency  $\omega_c$  is the *carrier frequency*, and controls the center frequency of the modulated signal. Viewed in another way, (6.45) allows us to map a baseband signal  $s(t)$  into a passband signal  $x(t)$ , a process called *modulation*. Section 2.4 also showed how to recover  $s(t)$  from  $x(t)$ , a process called *demodulation*. Usually we choose  $\omega_c$  large enough that  $s(t)e^{j\omega_c t}$  has no negative frequency components, and hence is *analytic*, as shown in Figure 6-13.

For digital communication,  $s(t)$  can be a baseband PAM signal, in which case  $x(t)$  will be called a passband PAM signal. Section 2.4 explained three modulation methods: AM-DSB, AM-SSB, and QAM. Of these three modulation techniques, QAM is preferred, and will be used here. AM-DSB is bandwidth-inefficient, because it forces the upper and lower sidebands to be conjugate-symmetric, or equivalently it requires that  $s(t)$  be real-valued. The conjugate symmetric sidebands are redundant, resulting in use of twice bandwidth necessary. AM-SSB and QAM have the same bandwidth efficiency, but AM-SSB is difficult to implement for baseband PAM waveforms (because it requires a phase splitter at baseband, which results in a frequency-domain discontinuity at d.c.).

In QAM, the baseband signal is allowed to be any complex-valued baseband signal; that is, unlike AM-DSB and AM-SSB, there is no enforced relationship between the real and imaginary parts of  $s(t)$ . Hence, the real and imaginary parts can both carry information. For this reason, the spectrum  $S(j\omega)$  shown in Figure 6-13 is not symmetric in general. In the baseband PAM signal,

$$s(t) = \sum_{k=-\infty}^{\infty} a_k g(t - kT) , \quad (6.46)$$

we can make the data symbols  $a_k$  complex-valued, or we can make the baseband



**Figure 6-13.** An example of a baseband signal (a), shown in the frequency domain, its analytic passband equivalent (b), and its real-valued passband equivalent (c). The relationship in the time domain is given by (6.45).

pulse  $g(t)$  complex-valued, or both. Making the data symbols complex-valued is particularly valuable, because it allows us to double the information transferred (we can think of this as transmitting two real-valued data symbols, the real part and the imaginary part). There is no particular motivation to make  $g(t)$  complex valued, so we will assume that it is real valued.

The modulation of (6.45) in combination with the complex-valued baseband PAM signal of (6.46) will be called *passband PAM*. The reason we do not use the terminology QAM that was used for the more general modulation context of Section 2.4 is that we reserve the term QAM for a specific choice of data symbol alphabet (as defined in Section 6.5). As considered in Section 6.5, a wide variety of modulation techniques used in digital communication are special cases of passband PAM.

A block diagram of a passband PAM modulator is shown in Figure 6-14. The bits are mapped by the coder into complex-valued data symbols  $a_k$ , and passed through a real-valued transmit filter  $g(t)$ . After modulation by  $e^{j\omega_c t}$ , the signal is analytic (having only positive frequency components), and the real part is a passband signal suitable for transmission over a passband channel. The  $\sqrt{2}$  factor ensures that the energy of the complex-baseband and passband signals are the same.

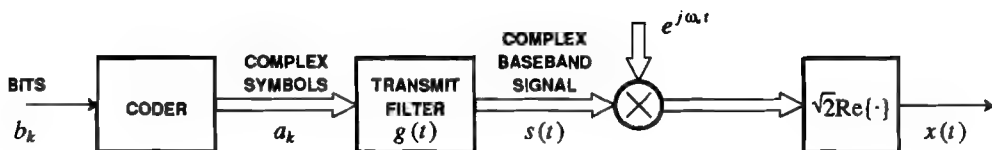
We can compare the bandwidth required for the channel in the baseband and passband cases. If the baseband signal has bandwidth  $W$ , then the passband PAM bandwidth is  $2W$ , because of the upper and lower sidebands, as is evident from Figure 6-13; that is, both positive and negative frequency components of  $s(t)$  are represented by positive frequencies in  $x(t)$ , doubling the bandwidth. However, since we can think of passband PAM as transmitting two baseband PAM signals, one for the real part and one for the imaginary, the overall bandwidth efficiency is the same.

### 6.4.2. Three Representations for Passband PAM

We have been using the following representation for the passband PAM transmitted signal in Figure 6-14:

$$x(t) = \sqrt{2} \operatorname{Re} \left\{ e^{j\omega_c t} \sum_{m=-\infty}^{\infty} a_m g(t - mT) \right\}. \quad (6.47)$$

If the transmitted pulse  $g(t)$  is real-valued we get a second representation,



**Figure 6-14.** A passband PAM modulator. The most important difference from a baseband PAM modulator is that the coder maps bits into complex-valued data symbols.

$$x(t) = \sqrt{2} \left[ \cos(\omega_c t) \sum_{m=-\infty}^{\infty} \operatorname{Re}\{a_m\} g(t - mT) \right] - \sqrt{2} \left[ \sin(\omega_c t) \sum_{m=-\infty}^{\infty} \operatorname{Im}\{a_m\} g(t - mT) \right]. \quad (6.48)$$

In practice,  $g(t)$  is almost always real-valued. Thus Figure 6-14 is equivalent to modulating two real-valued baseband PAM signals,

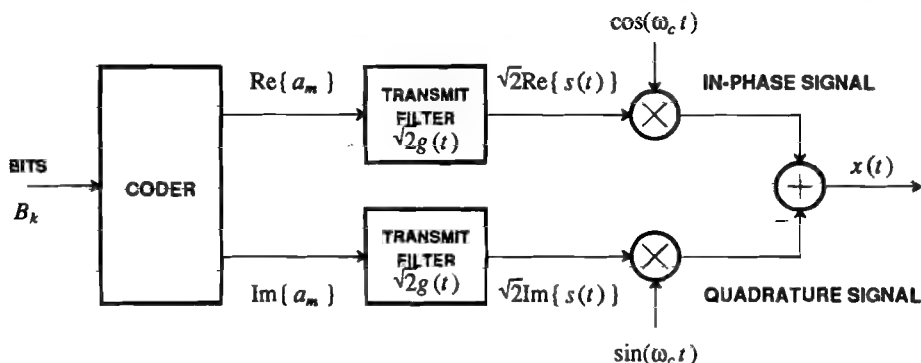
$$\sqrt{2} \sum_{m=-\infty}^{\infty} \operatorname{Re}\{a_m\} g(t - mT), \quad \sqrt{2} \sum_{m=-\infty}^{\infty} \operatorname{Im}\{a_m\} g(t - mT), \quad (6.49)$$

by the carrier signals  $\cos(\omega_c t)$  and  $-\sin(\omega_c t)$  respectively. These two carriers are 90 degrees out of phase with one another, so they are said to be in *quadrature*. The first term in (6.49), modulating the  $\cos(\omega_c t)$  carrier, is called the *in-phase component*, and the second term, modulating the  $\sin(\omega_c t)$ , is called the *quadrature component*.

### Example 6-21.

The representation (6.48) suggests a system in which  $\operatorname{Re}\{a_m\}$  and  $\operatorname{Im}\{a_m\}$  are selected independently from the same alphabet. We call this type of transmission quadrature amplitude modulation (QAM), and explore it further in Section 6.5. In the literature, the term QAM is often used to refer to any passband PAM signal, but we will use the term in a more restricted way.  $\square$

A realization of (6.48) is shown in Figure 6-15. While equivalent to Figure 6-14, it is obviously preferable for implementation because Figure 6-14 suggests that the imaginary part of  $s(t)e^{j\omega_c t}$  is computed, and then thrown away, while in Figure 6-15 the imaginary part is not computed. Nevertheless, in the remainder of this book we will tend to use the complex-valued notation of Figure 6-14 because it is much



**Figure 6-15.** A passband PAM transmitter. It performs the same function as the transmitter in Figure 6-14 when the transmit filter  $g(t)$  is real-valued.

more compact, and because it is easy to recognize situations where the computation of the imaginary part of a signal can be avoided.

A third representation of passband PAM follows by representing the data symbols  $a_m$  in terms of their magnitude and angle (polar coordinates),

$$a_m = c_m e^{j\theta_m} \quad (6.50)$$

so that

$$\begin{aligned} x(t) &= \sqrt{2} \operatorname{Re} \left\{ \sum_{m=-\infty}^{\infty} c_m e^{j(\omega_c t + \theta_m)} g(t - mT) \right\} \\ &= \sqrt{2} \sum_{m=-\infty}^{\infty} c_m \cos(\omega_c t + \theta_m) g(t - mT). \end{aligned} \quad (6.51)$$

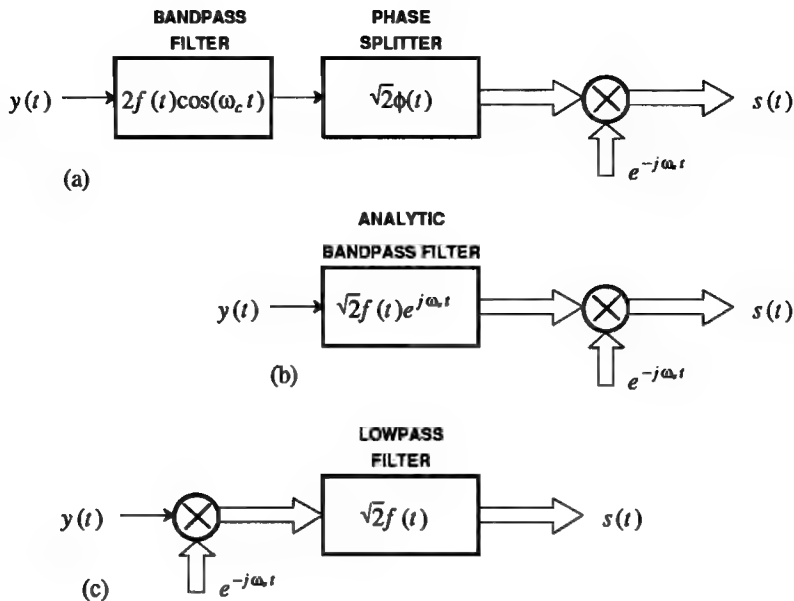
Each pulse  $g(t - mT)$  is multiplied by a carrier, where the amplitude and phase of the carrier is determined by the amplitude and angle of  $a_m$ . This is sometimes called AM/PM, for amplitude modulation and phase modulation. It suggests that *phase-shift keying (PSK)*, in which data is conveyed only on the phase of the carrier, is a special case of passband PAM. This is in fact true, and will be explored further in Section 6.5.

### 6.4.3. Passband PAM Receivers

A general demodulator structure which allows recovery of the complex-baseband signal  $s(t)$  from the passband signal  $x(t)$  was displayed in Figure 2-6. First the negative-frequency components are removed using a phase splitter, and then the positive-frequency components of the resulting analytic signal are demodulated to baseband. Unfortunately, a demodulator structure of that precise form is not practical for passband PAM, because it ignores the fact that there will typically be noise introduced in the channel. In addition, there will often even be other signals sharing the same channel with different carrier frequencies, as with radio channels (Chapter 5). In practice, there is the need for a *receive filter* to reject out-of-band noise and out-of-band signals, just as in the baseband case. In addition, the receive filter can compensate for frequency-dependent distortion on the channel, resulting in a pulse shape that satisfies the Nyquist criterion at the slicer input.

Accordingly, assume a baseband-equivalent receive filter  $f(t)$ , entirely analogous to the receive filter in the baseband case. An equivalent passband filter has impulse response  $2f(t)\cos(\omega_c t)$ . The normalization is such that the passband filter has the transfer function  $F(j(\omega - \omega_c)) + F(j(\omega + \omega_c))$ , which is the same transfer function shifted to passband. For example, if  $F(j\omega)$  is an ideal lowpass filter at baseband, then  $2f(t)\cos(\omega_c t)$  is an ideal bandpass filter. To emphasize the analogy to the baseband case, we propose to apply the bandpass filter  $2f(t)\cos(\omega_c t)$  to the received signal  $y(t)$  before demodulating. This is increasingly done in practice.

The resulting demodulator and two equivalent structures are shown in Figure 6-16. In Figure 6-16a we have simply added a passband real-valued receive filter before the demodulator of Figure 2-6. This structure has some advantages in important practical circumstances.



**Figure 6-16.** Three equivalent demodulator structures obtained from Figure 2-6 by adding a passband receive filter with equivalent baseband impulse response  $f(t)$ .

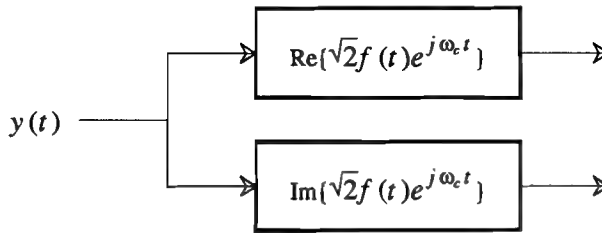
#### Example 6-22.

In a microwave radio system, the IF (intermediate frequency) bandpass filter that acts to reject other radio channels can also double as the receive filter in the configuration of Figure 6-16a. Since this filter is fairly expensive to realize using passive components, it is more desirable to realize a single filter rather than the two physical filters required for the analytic bandpass filter.  $\square$

#### Example 6-23.

In a voiceband data modem, the receive filter is often implemented in an analog front-end integrated circuit, and the phase splitter is realized in the discrete-time domain after sampling. The bandpass filter doubles as an anti-aliasing filter as well as a receive filter. The configuration of Figure 6-16a again offers the advantage of putting as much of the filtering as possible in discrete time.  $\square$

In Figure 6-16b we recognize that the receive filter and the phase splitter can be combined into a single filter. The resulting filter is still a passband filter, but it passes only positive frequency components, and not negative frequency components. Since the impulse response of this filter,  $\sqrt{2}f(t)e^{j\omega_c t}$ , is an analytic signal, we term this filter an *analytic passband filter*. The impulse response  $\sqrt{2}f(t)e^{j\omega_c t}$  is always complex-valued, so the filter will require two real filters for implementation, as shown below:



Finally, we display in Figure 6-16c a third structure, in which the receive filtering and demodulation are reversed. The equivalence of this structure is easily established by noting that

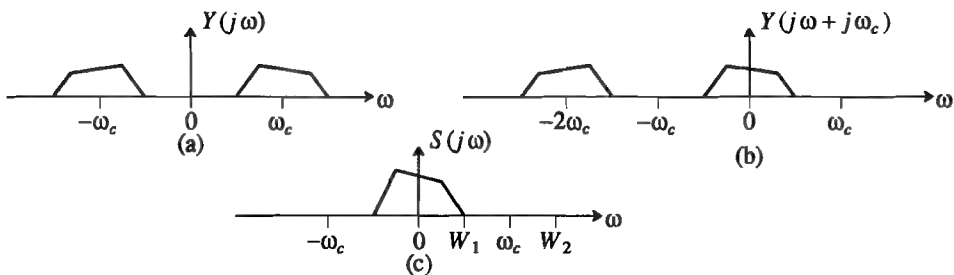
$$e^{-j\omega_c t} \int_{-\infty}^{\infty} y(\tau) \sqrt{2}f(t-\tau) e^{j\omega_c(t-\tau)} d\tau = \int_{-\infty}^{\infty} y(\tau) e^{-j\omega_c \tau} \sqrt{2}f(t-\tau) d\tau \quad (6.52)$$

or equivalently

$$e^{-j\omega_c t} \left[ y(t) * (\sqrt{2}f(t)e^{j\omega_c t}) \right] = (e^{-j\omega_c t} y(t)) * (\sqrt{2}f(t)). \quad (6.53)$$

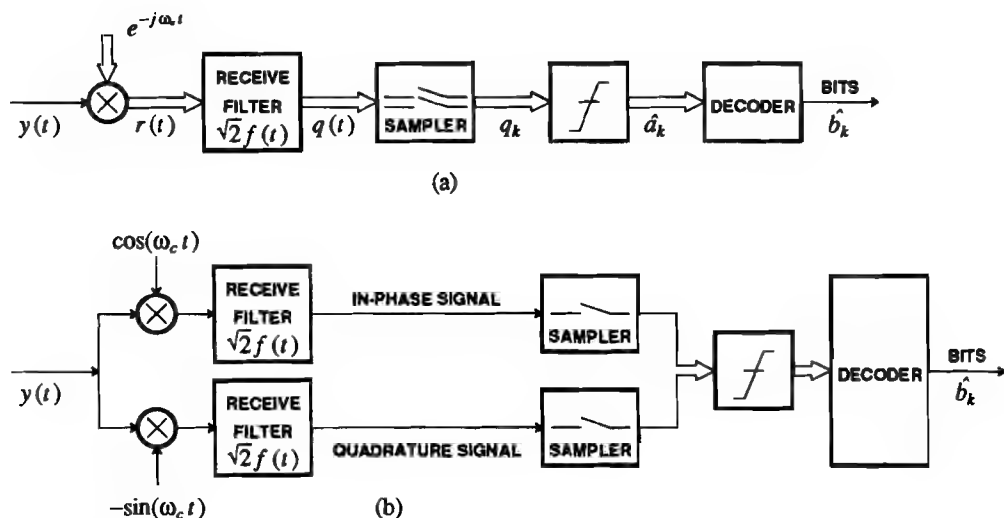
Intuitively, it performs the receive filtering function at baseband after first translating the received signal to baseband. The relevant signals are shown (in the frequency domain) in Figure 6-17. It also eliminates the double-carrier-frequency term that is absent in Figure 6-16a and b because the phase splitter eliminates negative-frequency terms before frequency translation.

Figure 6-16 shows only how to demodulate the received passband signal  $y(t)$  to recover the complex baseband signal  $s(t)$ , not how to detect the data symbols. The latter can be performed by sampling and slicing as in the baseband case, except that we now have a complex-valued PAM signal rather than real-valued. A complete passband PAM receiver, including both demodulation and detection, is shown in Figure 6-18 for the case of a demodulator preceding the baseband receive filter. Figure



**Figure 6-17.** Signals in Figure 6-16c are shown in the frequency domain. a. The received signal. b. After demodulation (note the double frequency components at  $-2\omega_c$ ). c. After lowpass filtering and scaling by  $\sqrt{2}$ .





**Figure 6-18.** A demodulator plus baseband receive filter structure for a passband PAM receiver. (a) In terms of complex-valued signals, and (b) the equivalent structure in terms of real-valued signals assuming the receive filter  $f(t)$  is real-valued. The detector structure is similar to the baseband case, except that the slicer is designed for complex-valued data symbols.

6-18b assumes that the receive filter  $f(t)$  is real-valued, which will often not be true as we will see shortly. However, this structure conveniently illustrates that the receiver can be thought of as two baseband PAM receivers operating in parallel, one using an in-phase carrier and the other a quadrature carrier.

A second receiver structure using an analytic passband filter prior to demodulation is shown in Figure 6-19. This structure illustrates a simplification that occurs when we combine the demodulator and baseband PAM detector. Since the symbol-rate sampler immediately follows the demodulator, the sampler and demodulator can be reversed. The demodulation is then performed in the discrete-time domain. This has important practical consequences, because it is common to coordinate the choice of symbol rate and carrier frequency at the transmitter so that the quantity  $\omega_c T$  assumes a convenient value. For example, if  $\omega_c T = 2\pi/N$ , then the values of  $e^{j\omega_c kT}$  can be easily generated from a lookup table with  $N$  entries. This is often simpler to implement than generating  $e^{j\omega_c t}$  and performing the multiplication in continuous time.

### Noise Power Spectrum at Receive Filter Output

As in the baseband case, assume that the channel introduces white Gaussian noise with power spectrum  $S_N(j\omega) = N_0$ . We can determine the power spectrum of the noise component at the output of the receive filter, again showing that, intuitively, the bandwidth of the receive filter should be made small. We will use the demodulator structure of Figure 6-16b. Assume, as shown in Figure 6-20, that the noise alone

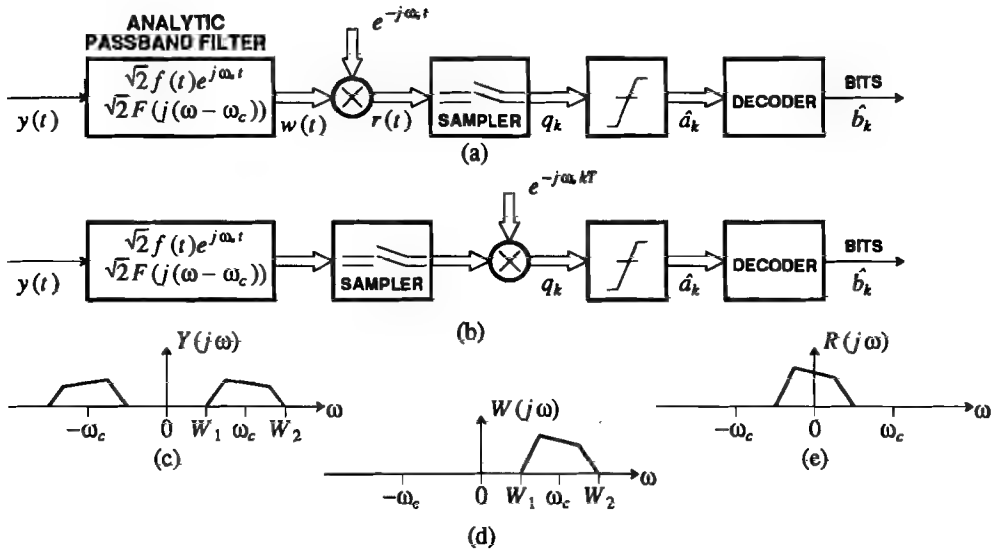


Figure 6-19. Two receivers equivalent to Figure 6-18 using an analytic passband filter. Also shown are the Fourier transforms of the deterministic received signal (c), the output of the analytic passband filter (d), and the output of the demodulator (e).

(no signal) is applied to the demodulator. Now let us determine the power spectrum of the baseband noise  $Z(t)$ . From Figure 6-20, the noise  $M(t)$  has power spectrum

$$S_M(j\omega) = 2N_0 |F(j(\omega - \omega_c))|^2 \tag{6.54}$$

**Exercise 6-2.**  
Show that

$$R_Z(\tau) = e^{-j\omega_c \tau} R_M(\tau) \tag{6.55}$$

and hence

$$S_Z(j\omega) = S_M(j(\omega + \omega_c)) = 2N_0 |F(j\omega)|^2 \tag{6.56}$$

□

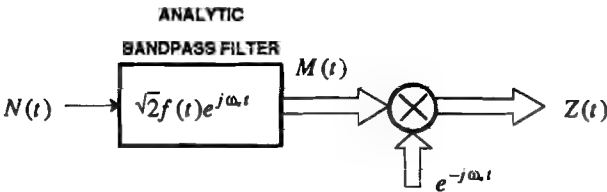


Figure 6-20. The demodulator of Figure 6-16b with noise only at its input.

It is not surprising that the power spectrum of the noise is proportional to the squared magnitude of the receive filter response. Again, this result suggests minimizing the bandwidth of the receive filter. We defer this topic until Chapter 8, where we examine the properties of this complex-valued noise in detail.

#### 6.4.4. Equivalent Baseband Representations

The received signal at the input to the receiver (output of the channel) can be written in a form similar to the transmitted passband PAM signal, except that the transmitted pulse shape  $g(t)$  is replaced by another pulse shape  $h(t)$  that takes into account the effect of the channel,

$$y(t) = \sqrt{2} \operatorname{Re} \left\{ e^{j\omega_c t} \sum_{k=-\infty}^{\infty} a_k h(t-kT) \right\}, \quad (6.57)$$

neglecting the noise. We call  $h(t)$  the *equivalent baseband pulse*. Again the  $\sqrt{2}$  factor ensures that the power of the passband signal is the same as that of the baseband signal.

##### Exercise 6-3.

Show that for a transmitted pulse  $g(t)$  and channel impulse response  $b(t)$ , the received baseband pulse has Fourier transform

$$H(j\omega) = B[j(\omega + \omega_c)]G(j\omega). \quad (6.58)$$

For  $\omega_c \neq 0$  this spectrum does not usually have conjugate symmetry about d.c., and hence  $h(t)$  is in general *complex-valued*.  $\square$

In the time domain, the equivalent baseband pulse can be written as

$$h(t) = b_E(t) * g(t), \quad (6.59)$$

where  $b_E(t)$  is the equivalent complex-valued baseband impulse response of the channel

$$b_E(t) = e^{-j\omega_c t} b(t). \quad (6.60)$$

The equivalent baseband transfer function of the channel is

$$B_E(j\omega) = B[j(\omega + \omega_c)], \quad (6.61)$$

and is therefore nothing more than the passband transfer function in the vicinity of the carrier frequency shifted down to d.c.

For some special cases, the equivalent baseband response  $h(t)$  is real-valued, as illustrated in the following examples.

##### Example 6-24.

When  $\omega_c = 0$ ,  $h(t)$  is real valued and  $e^{j\omega_c t} = 1$ , and thus baseband PAM reception is a special case of passband PAM reception.  $\square$

**Example 6-25.**

When the channel transfer function  $B(j\omega)$  is conjugate-symmetric about the carrier frequency,

$$B^*[j(\omega_c - \omega)] = B[j(\omega_c + \omega)] , \quad |\omega| < \omega_c , \quad (6.62)$$

and

$$B^*[j(-\omega_c - \omega)] = B[j(-\omega_c + \omega)] , \quad |\omega| < \omega_c , \quad (6.63)$$

then  $h(t)$  is real-valued. This could include the special case where the channel is an ideal bandpass filter centered at frequency  $\omega_c$ .  $\square$

**Receive Signal After Filtering and Demodulation**

Given the receive signal of (6.57), we can easily find the signal at the output of the receive filter and demodulator, given by

$$q(t) = \left[ \sum_{k=-\infty}^{\infty} a_k h(t-kT) \right] * f(t) , \quad (6.64)$$

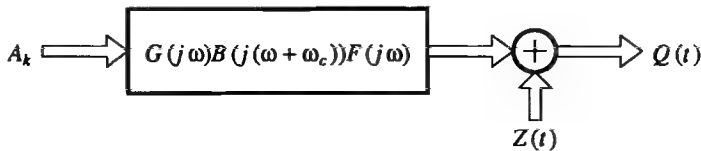
again neglecting the noise. The result is a baseband PAM signal with equivalent pulse shape  $p(t) = h(t) * f(t)$ . Even for a real-valued receive filter,  $p(t)$  is in general complex-valued.

**Equivalent Baseband Channel**

We can now give an equivalent baseband channel model, including the receive filter in the channel model, as shown in Figure 6-21. The equivalent channel filters the transmitted data symbols through a transfer function consisting of the product of

- The Fourier transform of the transmitted pulse,
- The transfer function of the channel in the vicinity of the carrier frequency shifted down to baseband, and
- The transfer function of the baseband receive filter.

In general this filter has a complex-valued impulse response. The additive noise  $Z(t)$  is the channel noise after filtering and demodulation. Its power spectrum is given by



**Figure 6-21.** Equivalent baseband channel model including modulator and demodulator, transmit and receive filter, and channel. The noise  $Z(t)$  is the channel noise after filtering and demodulation. Its power spectrum is given by (6.56). It will be further characterized in Chapter 8.

(6.56). It will be further characterized in Chapter 8, where we will show that it has jointly Gaussian real and imaginary parts which are independent and have the same variance when sampled at the same time  $t$ .

The pulse at the output of the receive filter is

$$P(j\omega) = F(j\omega)B_E(j\omega)G(j\omega), \quad (6.65)$$

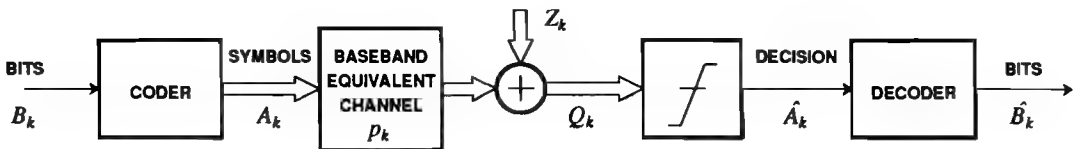
where  $B_E(j\omega) = B[j(\omega + \omega_c)]$  as given by (6.61). The only difference between this analysis and that in Section 6.3.3 is that these Fourier transforms can each correspond to complex-valued time-domain functions. To avoid ISI,  $P(j\omega)$  should satisfy the Nyquist criterion (6.23), which is the same for complex pulses as for real. A practical engineering approach (not necessarily optimal) is to choose a desired  $P(j\omega)$  satisfying the Nyquist criterion, and to design a complex-valued receive filter satisfying

$$F(j\omega) = \begin{cases} \frac{P(j\omega)}{B_E(j\omega)G(j\omega)}; & \text{for all } \omega \text{ such that } B_E(j\omega)G(j\omega) \neq 0 \\ 0; & \text{for all } \omega \text{ such that } B_E(j\omega)G(j\omega) = 0 \end{cases} \quad (6.66)$$

just as in (6.35). This approach has the significant disadvantage that if  $B_E(j\omega)G(j\omega)$  is small but non-zero for any range of  $\omega$  where  $P(j\omega)$  is not small, then  $F(j\omega)$  will have large gain in this range. This can result in significant amplification of the noise, a phenomenon known as *equalizer noise enhancement*. This phenomenon will be studied further in Chapter 10, where optimal receive filters are derived.

### 6.4.5. Equivalent Discrete-Time Representations

A passband PAM system can be modeled as a completely discrete-time system if the continuous-time subsystems are considered to be part of the channel. Such an interpretation is illustrated in Figure 6-22. In this case the transmit filter  $g(t)$  and receive filter  $f(t)$  must be included in the channel model. Assuming sampling at the symbol rate as in Figure 6-19, the equivalent discrete-time channel has impulse response  $p_k$  and transfer function  $P(z)$ , where



**Figure 6-22.** A passband communication system can be modeled using a complex-valued discrete-time channel.

$$P(e^{j\omega T}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} G[j(\omega - \frac{2\pi}{T}m)] B_E[j(\omega - \frac{2\pi}{T}m)] F[j(\omega - \frac{2\pi}{T}m)] \quad (6.67)$$

from (2.17). The equivalent noise  $Z_k = Z(kT)$  will be studied thoroughly in Chapter 8. Its power spectrum, however, is just an aliased version of (6.56),

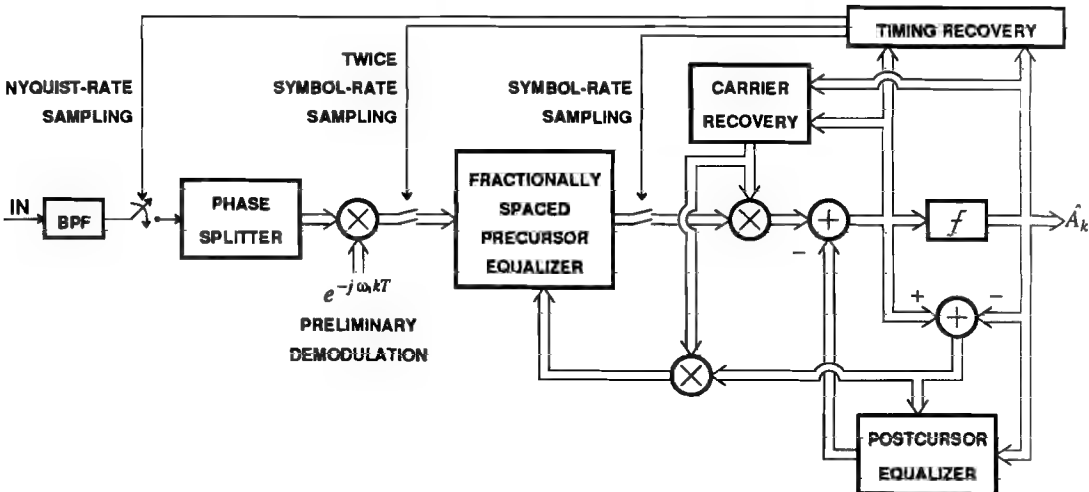
$$S_Z(e^{j\omega T}) = \frac{2N_0}{T} \sum_{m=-\infty}^{\infty} |F(j(\omega - m\frac{2\pi}{T}))|^2, \quad (6.68)$$

where  $T$  is the sample interval, which in Figure 6-22 equals the symbol interval.

The equivalent discrete-time channel model of Figure 6-22 will prove to be very useful since it abstracts all the details of the modulation, demodulation, and filtering into a single simple baseband model.

### 6.4.6. More Elaborate PAM Receivers: A Preview

The receivers that we have described in this section consist basically of a filter, a demodulator, and a slicer. The filter characteristics are derived from a common-sense requirement to reject out-of-band noise and to avoid ISI at the slicer. However, passband PAM receivers can be much more elaborate, as we will see in subsequent chapters. In order to motivate those chapters, we give here a qualitative description of a typical passband PAM receiver in Figure 6-23. It is a practical receiver, although there are many variations. The parts of the receiver that are already familiar are the bandpass filter on the front end, the phase splitter, the demodulator, and the slicer. In fact, the front end consisting of a BPF followed by a phase splitter is much like the structure shown in Figure 6-16a.



**Figure 6-23.** Block diagram of a typical passband PAM receiver. Specific parts of this receiver will be discussed in detail in subsequent chapters.

After reviewing some basics of detection theory in Chapter 9, we will derive the optimal receiver structure in Chapter 10. We will find that the front-end filtering and demodulation considered thus far in this chapter is optimal, as long as a particular filter transfer function called the *matched filter* is used. However, the optimal receiver uses much more complicated mechanisms for detecting the data symbols in the face of ISI. Careful compromises then lead to structures that look more promising and are made fully practical in Chapter 11. One such structure is a *decision-feedback equalizer* that consists of a *fractionally-spaced precursor equalizer* (often also called a "forward equalizer") and a *postcursor equalizer* (often called a "feedback equalizer"), as shown in Figure 6-23. The fractionally-spaced precursor equalizer is a filter that performs the function of the matched filter, and also equalizes the *precursor* portion of the ISI, which is defined as the interference from future data symbols. The postcursor equalizer then removes the *postcursor* portion of the ISI, defined as the interference from past data symbols. In Chapter 11 we show how the parameters of these filters can be *adapted* automatically so that characteristics of the channel do not have to be precisely known by the designer of the receiver.

*Timing recovery* is required to derive a symbol-rate clock from the PAM waveform itself, as shown in Figure 6-23 and explained in Chapter 17. There are many different timing recovery schemes available; the one shown here is *decision-directed*, which means that it uses the receiver decisions to update the phase and frequency of the clock. It is also shown producing three different sampling rates, all related by rational multiples. The Nyquist-rate sampling at the front end is required if the phase splitter is implemented in discrete time. Of course it need not be, and in fact can be combined with the bandpass filter at the front end, in which case this sampling operation will not be required. The second sampling rate is at twice the symbol rate; this explains the terminology "fractionally-spaced" for the subsequent equalizer. The final sampling operation is at the symbol rate, since the slicer requires samples only at the symbol rate.

There are also connections from the output of the slicer (the decisions) to the two equalizers. These connections are required for adaptation of the equalizers, and imply that adaptation is also decision-directed.

Also shown in Figure 6-23 is the *carrier recovery*, which will be explained in Chapter 16. Until Chapter 16 we will consistently assume that the precise carrier frequency and phase are available at the receiver (except for incoherent passband receivers in Section 6.8), but in practice this is not true. After the phase splitter in Figure 6-23, a preliminary demodulation is done using a carrier with frequency  $\omega_1$ . This carrier frequency is not expected to match the transmitter carrier frequency precisely, so phase errors result from the demodulation. These phase errors are corrected by further demodulation, shown as a complex multiplication following the fractionally-spaced precursor equalizer. The reason for this two-step demodulation is that the carrier recovery is decision-directed, like the timing recovery. A loop is formed that includes the slicer, the carrier recovery, and a complex multiplier, as shown in Figure 6-23. It will become clear in Chapter 16 that the performance of this structure is considerably improved if there is no additional filtering inside the loop (the postcursor equalizer is harmless in this configuration). Consequently the final

demodulation should be done as close to the slicer as possible. The preliminary demodulation, however, is required in order to bring the signal down close to baseband so that the receiver does not have to operate on the high frequency signal. Sometimes this first demodulation can be performed simply by sampling the signal below the Nyquist rate, without using the complex multiplier shown in Figure 6-23.

Some possible variations on the receiver shown in Figure 6-23 include the use of error correcting codes (Chapter 13) or trellis codes (Chapter 14), the use of a Viterbi detector instead of the slicer and equalizers, or the omission of the postcursor equalizer (Chapters 10 and 11). It is also practical to design passband signals that are not PAM signals, for example FSK (below) or *continuous-phase modulation*, in which case the receivers are significantly different. Baseband receivers can also be more elaborate than those discussed in Section 6.3 above, using for example *line coding* (Chapter 12) and adaptive equalization (Chapters 10 and 11).

## 6.5. ALPHABET DESIGN

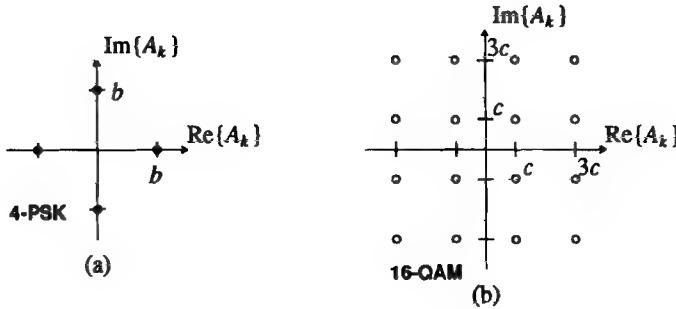
Having determined the equivalent baseband and discrete-time channels, we can now address the problem of designing the data symbol alphabet. For the purposes of this section, the entire system may be viewed as a discrete-time system as shown in Figure 6-22. A baseband communication system is just a special case where the symbols  $A_k$ , baseband equivalent channel  $p_k$ , and the noise  $Z_k$  are real-valued.

### 6.5.1. Constellations

The *alphabet* is the set of symbols that are available for transmission. The receiver uses a *slicer* which makes the decision about the intended symbol. The input to the slicer is a discrete-time signal with sampling interval equal to the symbol rate. When there is no *intersymbol interference* (ISI), then each sample into the slicer is equal to the transmitted data symbol corrupted by an additive noise that is independent of the symbol sequence. For the receivers considered so far, the noise component of the slicer input sample is Gaussian when the channel noise  $N(t)$  is Gaussian, as will be shown in Chapter 8. For our purposes here, we will consider the effect of the noise only at an intuitive level.

A baseband signal has a real-valued alphabet that is simply a set of real numbers, for example  $A = \{-3, -1, +1, +3\}$ . A passband PAM signal has an alphabet that is a list of complex numbers, for example  $A = \{-1, -j, +1, +j\}$ . Both of these example alphabets have size  $M = 4$ ; each symbol can represent  $\log_2 M = 2$  bits. A complex-valued alphabet is best described by plotting the alphabet as a set of points in a complex plane. Such a plot is called a *signal constellation*. There is a one-to-one correspondence between the points in the constellation and the signal alphabet. Two popular constellations are illustrated in the following examples.





**Figure 6-24.** Two popular constellations for passband PAM transmission. The constants  $b$  and  $c$  affect the power of the transmitted signal.

**Example 6-26.**

The 4-PSK constellation is shown in Figure 6-24a. It consists of four symbols of magnitude  $b$ , each with a different phase. Hence the symbols may be written

$$A_m = be^{j\phi_m} \quad (6.69)$$

and the transmitted signal may be written (from (6.51))

$$X(t) = b\sqrt{2} \sum_{m=-\infty}^{\infty} \cos(\omega_c t + \phi_m) g(t - mT) \quad (6.70)$$

where  $\phi_m$  assumes the four values from the set  $\{0, \pi/2, \pi, 3\pi/2\}$ . The information is carried on the phase of the carrier, while the amplitude of the carrier is constant, which explains the term *phase-shift keying* (PSK). The 4-PSK constellation is also called *quadrature phase-shift keying* (QPSK).  $\square$

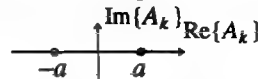
**Example 6-27.**

The 16-QAM constellation shown in Figure 6-24b has 12 possible phases and three amplitudes. Because of the rectangular nature of the constellation, the rectangular coordinate system is preferable to the polar coordinates that are natural for PSK.  $\square$

In a baseband PAM system, the real-valued alphabet can also be plotted as a one-dimensional constellation, although this is perhaps less informative.

**Example 6-28.**

The binary signal constellation below corresponds to a binary baseband PAM system:



This is called a *binary antipodal* signal constellation. It can be used for passband as well as baseband signaling, in which case it is sometimes called *binary phase-shift keying* (BPSK).  $\square$

Since baseband PAM is a special case of passband PAM, we will concentrate on passband PAM for the remainder of this chapter.

Because of additive noise and signal distortion, the received samples at the input to the slicer will not correspond exactly to points in the signal constellation, but if the noise power is small compared to the signal power, they will be close with high probability. If the noise at the slicer is Gaussian, the received samples will form a Gaussian cloud around the points in the constellation, as shown in Figure 6-25. This figure what we would see on an oscilloscope if we looked at a scatter plot of the discrete-time slicer input signal, with the horizontal axis being the real part and the vertical axis being the imaginary part.

Intuitively, the slicer should simply decide on the symbol in the constellation that is closest to received sample. In fact, when Gaussian noise is the only signal degradation, this is optimal in the *maximum likelihood* (ML) sense, as shown in Chapter 9. In that chapter, both the ML and *maximum a-posteriori* (MAP) detectors are shown to be slicers, and the decision thresholds are found.

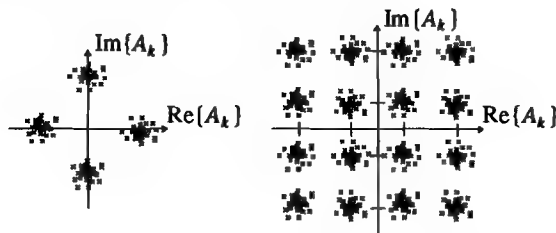
For both complex-valued and real-valued symbols, the intuitive slicer (which is also the ML detector) selects the  $\hat{A}_k$  in the alphabet that minimizes

$$|Q_k - \hat{A}_k|^2. \quad (6.71)$$

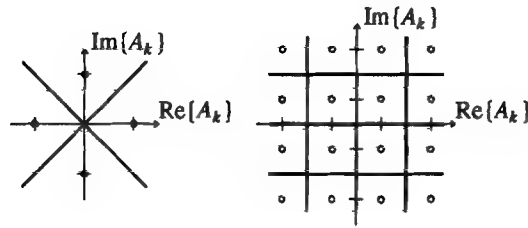
The slicer selects the symbol in the alphabet closest in Euclidean distance to the received signal sample. The complex plane can therefore be divided into *decision regions* where each decision region is the set of points that is closest to some symbol. The decision regions for the constellations in Figure 6-24 are shown in Figure 6-26.

### Minimum Distance

We would now like to be able to compare alphabets to determine which are best. A rigorous development, based on probability of error, is deferred to Chapter 8. But we can use intuition, based on Figure 6-25, to reach a remarkably simple conclusion. From Figure 6-25 it is clear that the distance between points in the constellation determines the likelihood that one point will be confused with another. Furthermore, two points are more likely to be confused for one another if they are closer together than if they are farther apart. Hence the *minimum distance* between points in the constellation, denoted  $d_{\min}$ , is a key parameter of the constellation.



**Figure 6-25.** Received samples perturbed by additive Gaussian noise form a Gaussian cloud around each of the points in the signal constellation.



**Figure 6-26.** The ML detectors for the constellations in Figure 6-24 have the decision regions shown.

Two constellations can be considered to have approximately the same noise immunity if the minimum distance  $d_{\min}$  is the same. But to make  $d_{\min}$  the same, constellations with more points (such as 16-QAM vs. 4-PSK) require more transmit power. Hence there is either a power or an error penalty associated with using larger constellations.

### Power Constraints

Practical channels impose a constraint on the average or peak power of the transmitted signal.

#### Example 6-29.

On the telephone channel, the average transmitted power is constrained by regulation. This is done so that the voiceband data signal will be comparable in power to voice signals. Many long distance facilities, particularly of the analog variety, are carefully designed under assumptions on the average power of each voiceband channel. On radio channels the average power is often constrained by regulation to avoid interference with other radio services. In addition, nonlinearities in the RF circuitry become more severe as the signal power gets larger. In wire-pair channels, the signal power is constrained so as to limit crosstalk interference with other cable pairs.  $\square$

We will now show that an average transmitted power constraint is equivalent to a constraint on the average of the squared magnitude of the transmitted symbols. Assume that the symbol sequence is white, so that the power spectrum is a constant:

$$S_A(e^{j\omega T}) = \sigma_A^2. \quad (6.72)$$

This assumption is usually valid (for exceptions, see Chapter 12). From Appendix 3-A, the power spectrum of the transmitted signal  $S(t)$  is

$$S_S(j\omega) = \frac{1}{T} |G(j\omega)|^2 S_A(e^{j\omega T}) = \frac{\sigma_A^2}{T} |G(j\omega)|^2 \quad (6.73)$$

where  $G(j\omega)$  is the transfer function of the transmit filter. The power of the complex-valued baseband PAM signal  $S(t)$  is

$$P_S = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_S(j\omega) d\omega. \quad (6.74)$$

In the passband case we transmit

$$X(t) = \sqrt{2} \operatorname{Re}\{ e^{j\omega_c t} S(t) \} \quad (6.75)$$

which has the same power as  $S(t)$ ,  $P_X = P_S$ . Let the energy in the transmit pulse be written as

$$\sigma_g^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega \quad (6.76)$$

so that the transmitted power can be written as

$$P_X = \frac{1}{T} \sigma_A^2 \sigma_g^2. \quad (6.77)$$

If the transmitted signal is constrained to power  $P$ , then the signal constellation should be constrained so that the expected magnitude squared does not exceed  $PT/\sigma_g^2$ . This effectively sets an upper bound on the minimum distance  $d_{\min}$ , for any given constellation design.

### Constellation Design

Intuitively, the objective of signal constellation design is to maximize the distance between symbols while not exceeding the power constraint. This will increase the immunity to noise, as can be seen from Figure 6-25. Optimal constellations are often difficult to derive, and modems that use them may be unnecessarily costly. In this section, we describe some popular constellations that have close to optimal performance. We assume an average power constraint, but the results are easily extended to a peak power constraint.

Since the performance of a constellation depends only on the distances among symbols, we expect the performance of a constellation to be invariant under translation. Hence we should translate a constellation so that its power is minimized. We now show that its power will be minimized if it has zero mean. In other words, given a set of symbols  $\{a_i\}$ , we wish to translate them with a complex number  $m$  such that the power

$$E[|A - m|^2] = \sum_{i=1}^M p_A(a_i) |a_i - m|^2 \quad (6.78)$$

of the translated symbol set  $\{a_i - m\}$  is minimized. Note that (6.78) is precisely the expression for the moment of inertia of  $M$  point masses, where the mass of the  $i^{\text{th}}$  point is  $p_A(a_i)$  and its position is  $(a_i - m)$  [3]. This is easily shown to be minimized if  $m$  is taken to be the centroid (center of gravity) of the untranslated point masses; in other words, translate the system so that the centroid is at the origin. Thus the best choice for a translation is

$$m = E[A]. \quad (6.79)$$

To prove this, note that for any other translation  $n$ ,

$$\begin{aligned} E[|A - n|^2] &= E[|(A - m) + (m - n)|^2] \\ &= E[|A - m|^2] + 2(m - n)(E[A] - m) + |m - n|^2 \quad (6.80) \\ &= E[|A - m|^2] + |m - n|^2 \end{aligned}$$

where the last step follows from (6.79). The mean energy under the translation  $n$  is larger than the mean energy under the translation  $m$  by the amount  $|m - n|^2$ . From this it is clear that alphabets with zero mean always perform better than translated alphabets with non-zero mean under an average power constraint.

Aside from ensuring zero mean (which is easy), the problem of optimal design of the constellation is complicated. A group of theoretical papers in the early 1960's [4,5,6,7] developed some basic design techniques. We will concentrate on constellations that are used in practice.

Some *quadrature amplitude modulation* (QAM) constellations are shown in Figure 6-27. The constellations are classified according to the number of bits  $N$  per symbol that they can convey. The number of points in the constellation is therefore  $M = 2^N$ . We have restricted  $N$  to be even so that half the bits are represented by the value along the imaginary axis and half by the value along the real axis. There are two significant practical advantages to these types of constellations. First, the in-phase and quadrature signals are independent  $(N/2)$  level PAM signals, so the design of the coder is simple. Second, because of the regular rectangular pattern, the decision regions are defined by thresholds along the real and imaginary axes, so the decision (made by the slicer) is easy. There is a price for this convenience. Rectangular constellations are not the most efficient in power for a given minimum distance between symbols.

Constellations for odd  $N$  are also possible (and practical), as shown in Figure 6-28. The  $N = 1$  constellation is the familiar binary antipodal signal constellation. Recall that since the symbols are real-valued, this signal can be transmitted in baseband, if a baseband channel is available. Because of their shape, the  $N = 5$  and  $N = 7$  ( $M = 32$  and  $M = 128$ ) constellations are called *cross constellations*. These

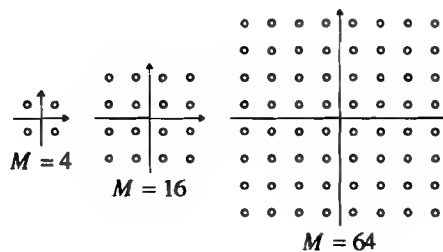


Figure 6-27. Some QAM constellations.

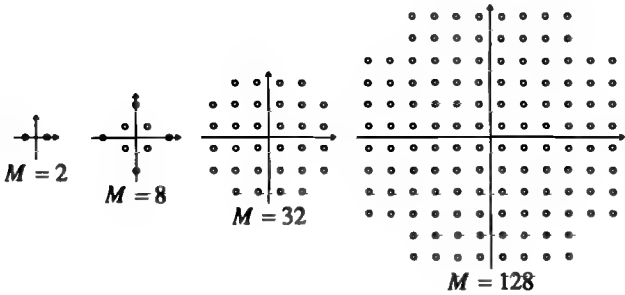


Figure 6-28. Cross constellations.

constellations require slightly more complicated coders than the QAM constellations and are often used in combination with *trellis coding* (see Chapter 14 and Problem 6-18). Both the QAM and the cross constellations are called *rectangular constellations* because the symbols are on a rectangular lattice.

Another family of constellations with simple geometry is based on phase-shift keying, sometimes combined with amplitude modulation, shown in Figure 6-29. The 2-PSK signal is identical to the binary antipodal signal, and is sometimes called *binary phase-shift keying* (BPSK). The 4-PSK signal, which we saw in Example 6-26, is sometimes called QPSK or 4-QAM.

Traditionally, pure PSK (without any amplitude modulation) has been used because inexpensive receivers can be designed with digital logic and because the signal has a constant envelope. The reason for this is that all the information about the signal is borne by the phase of the carrier, so the received signal can immediately be hard limited. Hard limiting preserves the locations of the zero crossings, which indicate the phase, and the hard-limited signal can be processed with digital logic without requiring A/D conversion.

A performance improvement can be achieved with hexagonal constellations, some examples of which are shown in Figure 6-30. The symbols lie on the vertices of equilateral triangles. The term *hexagonal* refers to the shape of the decision regions, shown for  $M = 16$  in Figure 6-30. For large  $M$ , hexagonal constellations minimize the extent of a constellation for a given distance between points (try penny packing),

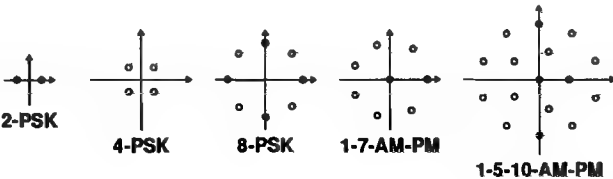
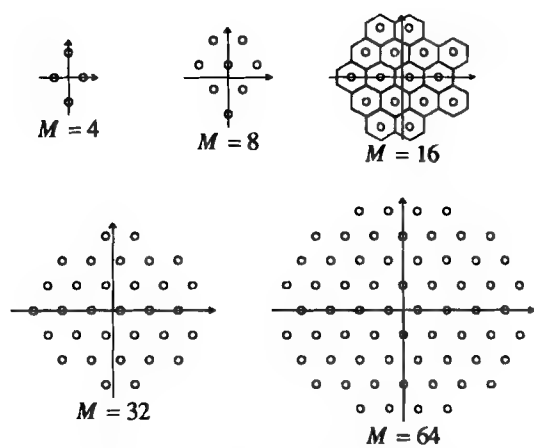


Figure 6-29. Constellations using phase-shift keying and amplitude modulation.



**Figure 6-30.** Optimal hexagonal constellations. For the  $M = 16$  constellation we have shown the hexagonal decision regions. The outer decision regions are approximated as hexagonal for uniformity.

and were suggested very early [8]. However, the improvement over rectangular constellations is slight, and the coder and slicer are significantly more complicated [9,10].

A systematic method for constructing optimal constellations is given in [9], but because of the limited benefit and serious complication of the detector these more elaborate constellations have not found widespread use. Constellations where the probability of the symbols is not uniform have also been proposed, but practical difficulties often dominate. Some performance gains can also be accomplished using constellations of higher dimension, such as four or eight. Essentially, several symbols are selected by the coder simultaneously and transmitted sequentially [11] or simultaneously on different carriers (Section 6.8 below).

**6.5.2. Differential Encoding and DPSK**

Most of the signal constellations we have described have the practical problem that they are rotationally invariant for some particular angles of rotation, typically multiples of  $\pi/2$ . By this we mean that if the constellation is rotated, there is no way that the receiver can distinguish it from a valid constellation, unless it knows the actual transmitted data symbols, which it does not. If this problem is not dealt with, the receiver may decide on a receive phase corresponding to a rotated constellation, with the result that the information bits will be incorrectly decoded. This problem can be eliminated by using *differential encoding*, in which the information is encoded by the *change* in constellation position between symbols rather than absolute position. Differential encoding can be mixed with absolute encoding. For the 16-QAM constellation of Figure 6-27, for example, it is common to differentially encode the quadrant (specified by two bits), while the point within the quadrant (specified by another two bits) is encoded absolutely.

Differential encoding is especially valuable on channels with rapid fading, such as the mobile radio channel described in Section 5.4. It is common on such channels to use PSK modulation, which makes the detection of data symbols insensitive to fluctuations in the amplitude of the received signal, since the slicer considers only the angle and not the amplitude of its input. However, as we also saw in Section 5.4, the phase of such a channel can also vary rapidly, especially during deep amplitude fades. At the expense of some noise immunity, this phase fluctuation can be mitigated by using PSK with differential encoding and differential detection. This combination of PSK, differential encoding, and differential decoding is called *differential PSK (DPSK)*.

We will consider differential encoding only for the case of PSK, although it should be recognized that it is often used with other constellations as well. For PSK, the transmitted symbols are of the form  $A_k = e^{j\phi_k}$  where the phases  $\phi_k$  are chosen from some alphabet. The phase  $\phi_k$  is determined by

$$\phi_k = \phi_{k-1} + \Delta_k, \quad (6.81)$$

where the difference in phase from one symbol to the next,  $\Delta_k$ , carries the information, not the absolute phase  $\phi_k$ .

**Example 6-30.**

In *differential binary PSK (DBPSK)*, one of two phases is transmitted. For this case, these two phases are  $\pi$  apart, and the coder can map a zero bit into  $\Delta_k = 0$  (two successive transmitted phases are identical) and a one bit into  $\Delta_k = \pi$  (two successive transmitted phases are  $\pi$  apart).  $\square$

**Example 6-31.**

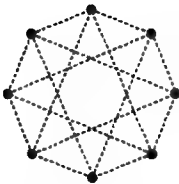
The IS-54 standard for digital cellular radio in North America transmits two bits per symbol, using is a form of *quadrature PSK (QPSK)*. However, rather than associating these two bits with four phases, in actuality eight equally-spaced phases are used, as shown in Figure 6-31. At any given symbol ( $k$ ) the data symbol assumes only one of four phases chosen from the sets  $\{0, \pi/2, \pi, 3\pi/2\}$  (for odd-numbered symbols) and  $\{\pi/4, 3\pi/4, 5\pi/4, 7\pi/4\}$  (for even-numbered symbols), where these two sets are offset by  $\pi/4$  relative to one another. Two information bits are coded as a *change* in phase by one of the values  $\{\pi/4, 3\pi/4, 5\pi/4, 7\pi/4\}$ . The possible phase transitions from one symbol to another are shown in Figure 6-31. The differential phase  $\Delta_k$  is determined from the two input information bits in accordance with the following table:

Bit 1	Bit 2	$\Delta_k$
1	1	$5\pi/4$
0	1	$3\pi/4$
0	0	$\pi/4$
1	0	$7\pi/4$

$\square$

There are two choices in the receiver design when using differential encoding: *coherent* or *synchrodyne detection*, which attempts to learn and track the absolute



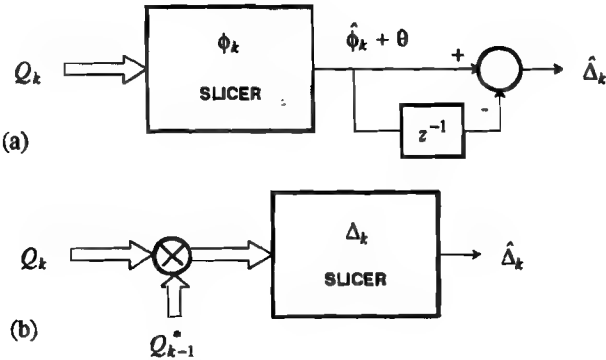


**Figure 6-31.** The North American IS-54 digital cellular standard uses eight phases to transmit two bits of information. The two bits are mapped into one of four phase transitions from one symbol to the next. These transitions are shown as dashed lines for each starting phase.

phase of the received data symbols, and *differential detection*, which looks at only the change in phase from one symbol to the other, as illustrated in Figure 6-32. Synchrony detection (Figure 6-32a) is appropriate when differential encoding is used only to mitigate the rotational invariance of the signal constellation. Suppose the input samples to the detector in both cases are

$$Q_k = e^{j\phi_k + \theta_k} + Z_k \tag{6.82}$$

where  $\theta_k$  is some unknown phase rotation and  $Z_k$  is the complex-valued additive noise. In the coherent case, we assume that  $\theta_k = \theta$ , where  $\theta$  assumes certain discrete phases (e.g. any multiple of  $\pi/4$  in Example 6-31) that allow the slicer to work properly. The  $Q_k$  are applied to a conventional slicer designed for the  $\phi_k$ , and it is assumed that the receiver estimates  $\phi_k + \theta$  rather than  $\phi_k$ . After the slicer, a difference operation forms an estimate of  $\Delta_k$ , independent of  $\theta$ , that directly represents the information bits.



**Figure 6-32.** Two detection techniques for DPSK. (a) Coherent, which requires an accurate phase reference, and (b) differential, which allows an arbitrary slowly-varying phase rotation of the data symbols.

The second alternative, differential detection, is shown in Figure 6-32b. This approach avoids tracking a rapidly varying channel phase. For this case, the statistic  $Q_k Q_{k-1}^*$  is formed before the slicer. The slicer is designed to have the proper thresholds for  $e^{j\Delta_k}$  rather than  $e^{j\Phi_k}$ . There are two consequences of this:

- In the absence of noise, the input to the slicer is the proper phase  $\Delta_k$  regardless of  $\theta$ . This is valuable on channels with rapid phase variations, since it means that the carrier phase does not have to be tracked.
- There is an increase in the noise at the slicer input; this is the price paid for the insensitivity to phase rotation.

We will now verify these two properties. The slicer input is

$$Q_k Q_{k-1}^* = e^{j\Delta_k} e^{j(\theta_k - \theta_{k-1})} + e^{j(\phi_k + \theta_k)} Z_{k-1}^* + e^{-j(\phi_{k-1} + \theta_{k-1})} Z_k + Z_k Z_{k-1}^* . \quad (6.83)$$

Assume that the phase rotation  $\theta_k$  does not change too much from one symbol to the next ( $\theta_k \approx \theta_{k-1}$ ). This is a valid assumption as long as the symbol rate is high relative to the rate of phase change. With this assumption, the signal term at the slicer input is  $e^{j\Delta_k}$ , independent of  $\theta_k$ . Looking at the noise terms,  $Z_k Z_{k-1}^*$  is the product of two noise terms, and hence will typically be insignificant. The phase factors multiplying  $Z_k$  and  $Z_{k-1}^*$  do not affect their variance. Approximating these terms as independent, the total noise variance is now

$$E[|Z_k + Z_{k-1}^*|^2] = 4\sigma^2 , \quad (6.84)$$

or twice as large as in the coherent case. There is thus roughly a 3 dB penalty for differential detection (twice as much noise power). A more refined analysis that takes account of the correlation of the two noise terms reveals that the penalty is actually about 2.3 dB at high SNR.

### 6.5.3. Spectral Efficiency

Recall that spectral efficiency, defined in (6.7), is a measure of the bit rate achieved per Hz of bandwidth. To determine the maximum achievable spectral efficiency for QAM, we use (6.8), repeated here for convenience,

$$\nu = \frac{\log_2 M}{BT} , \quad (6.85)$$

where  $M = |\Omega_A|$  is the alphabet size,  $B$  is the bandwidth in Hz, and  $T$  is the symbol interval. In section 6.2.1, we showed that the minimum bandwidth signal that satisfies the Nyquist criterion has bandwidth  $W = \pi/T$  radians/sec, or  $B = 1/2T$  Hz (see (6.23)). Thus, for a minimum bandwidth baseband PAM signal that avoids ISI,  $BT = 1/2$ , so

$$\nu = 2 \cdot \log_2 M \text{ bits/sec-Hz} . \quad (6.86)$$

Any higher spectral efficiency would imply ISI. For passband PAM, the same minimum pulse bandwidth is required, but the modulated signal will occupy twice the channel bandwidth (see Figure 6-13). Thus, the best spectral efficiency for a passband PAM signal that avoids ISI is

$$\nu = \log_2 M \text{ bits/sec-Hz} . \quad (6.87)$$

In both cases, the pulse shape required to achieve this spectral efficiency is impractical, so this bound is not achievable in practice.

Lest the reader infer that passband PAM has lower spectral efficiency than baseband PAM, recall that the alphabet can have more symbols in the passband case without significantly compromising performance. In fact, if we use QAM and transmit  $N$  levels on each of two quadrature carriers, the spectral efficiency is

$$\nu = \log_2 N^2 = 2 \log_2 N \text{ bits/sec-Hz} , \quad (6.88)$$

the same as for the baseband system with  $N$  levels. So the efficiency of baseband and passband PAM are effectively identical, other considerations being equal.

#### Example 6-32.

To achieve 4.5 bits/sec-Hz in a digital radio system (6.87) implies an alphabet size of at least  $M = 23$ , but considering the need for some excess bandwidth, and the convenience implied if  $M$  is power of two,  $M$  will be larger in practice. Let  $B$  be the spacing between carriers in a frequency-division-multiplexed digital radio system. Then the nominal bandwidth available on each carrier is  $B$  and a zero excess bandwidth system would have a symbol rate of  $1/T = B$ . In fact, the FCC transmission mask (Section 5.4) can be met for a digital radio system with  $1/T = (3/4)B$  and raised-cosine shaping with  $\alpha = 0.5$  [1]. The signal bandwidth is therefore  $3/2 \cdot 1/T = 9/8 \cdot B$  or 12.5% larger than the available bandwidth. This is acceptable, since the resulting interference with the adjacent carrier is small (the band edges of the raised-cosine pulse are small enough). The resulting spectral efficiency is

$$\nu = \frac{\log_2 M}{BT} = \frac{3}{4} \log_2 M \quad (6.89)$$

and 4.5 bits/sec-Hz can be achieved with  $M = 64$ . Thus, the number of points in the constellation is more than twice as great as with zero excess bandwidth. This is the price paid for practical filtering characteristics and tolerance for timing errors (Chapter 17).  $\square$

## 6.6. THE MATCHED FILTER — ISOLATED PULSE CASE

In Sections 6.3 and 6.4 we derived receiver structures for PAM without fully specifying the receive filter  $f(r)$ . It was argued that in order to eliminate ISI at the slicer input, the receive filter should be designed to yield a pulse satisfying the Nyquist criterion at the slicer. There are two problems with this that must be addressed:

- The Nyquist criterion does not uniquely specify the pulse, and hence the receive filter. Within the degrees of freedom available, we would like to choose a receive filter that maximizes the signal to noise ratio.
- At this point, we have no indication that the receiver structure assumed in Sections 6.3 and 6.4 is the best possible.

The full answers to both questions will have to await further developments in Chapters 7 through 10, where optimal receiver structures are derived in the presence of ISI.

We can move one step closer to answering these questions here if we arbitrarily eliminate any ISI considerations by assuming that only one pulse is transmitted, and design the receive filter to maximize the signal-to-noise ratio at the slicer. This is called the *isolated pulse* case. We will derive the *matched-filter receiver* and then show that it is equivalent to the *correlation receiver*.

### 6.6.1. Baseband Case

ISI can be ignored if we transmit a single data symbol  $A_0$ . Then the received signal in the baseband case is

$$Y(t) = A_0 \cdot h(t) + N(t) \quad (6.90)$$

where  $h(t)$  is the real-valued received pulse shape and  $N(t)$  is additive white Gaussian noise. We will assume a receiver structure similar to Section 6.2, consisting of a real-valued receive filter  $f(t)$ , followed by a sampler at  $t = 0$  and a slicer, except that now we will be able to unambiguously optimize the receive filter since we do not have to be concerned about ISI. The slicer input is

$$Q_0 = \int_{-\infty}^{\infty} Y(\tau) f(t - \tau) d\tau \Big|_{t=0} = \int_{-\infty}^{\infty} Y(\tau) f(-\tau) d\tau. \quad (6.91)$$

Note that the slicer input is of the form of a cross-correlation of the received signal with the time-reversed receive filter impulse response. Substituting (6.90) for  $Y(t)$ ,

$$Q_0 = A_0 \int_{-\infty}^{\infty} h(\tau) f(-\tau) d\tau + \int_{-\infty}^{\infty} N(\tau) f(-\tau) d\tau. \quad (6.92)$$

The first term is the signal and the second term is the noise. The variance of the noise term is easily shown to be

$$\sigma^2 = N_0 \int_{-\infty}^{\infty} f^2(-\tau) d\tau. \quad (6.93)$$

Intuitively, making the first term (the signal term) in (6.92) larger while keeping the second term (the noise term) constant should improve performance. Assume that power constraints on the channel prevent us from accomplishing this by either increasing the magnitude of  $h(t)$  or increasing the symbol spacing in the alphabet. Thus, assume  $h(t)$  and  $A_0$  in (6.92) cannot be changed, and select the filter  $f(t)$  that maximizes the power of the first term relative to the second. Let

$$\sigma_A^2 = E[|A_0|^2]. \quad (6.94)$$

and define the *signal to noise ratio* to be

$$\text{SNR}_0 = \frac{\sigma_A^2 \left[ \int_{-\infty}^{\infty} h(\tau) f(-\tau) d\tau \right]^2}{\sigma^2} \quad (6.95)$$

We can uniquely choose the receive filter  $f(t)$  to maximize (6.95). To do this, we need the integral form of the *Schwarz inequality*, given in vector form in Section 2.6. For any two (possibly complex) integrable functions  $f_1(x)$  and  $f_2(x)$ ,

$$\left| \int_a^b f_1(x) f_2^*(x) dx \right|^2 \leq \left[ \int_a^b |f_1(x)|^2 dx \right] \left[ \int_a^b |f_2(x)|^2 dx \right] \quad (6.96)$$

with equality if and only if  $f_2(x) = K f_1(x)$  for some constant  $K$  [12]. This is actually the same as the Schwarz inequality of Section 2.6, but using the  $L_2$  inner product. Since everything in (6.95) is real-valued (we are considering the baseband case only at this time), the SNR is maximized if

$$f(t) = K h(-t) \quad (6.97)$$

for some constant  $K$ . This choice of receive filter is called the *matched filter*. Using this in (6.93) and (6.95), we can express the *matched-filter bound* on the SNR as

$$\text{SNR}_0 \leq \frac{\sigma_A^2 \sigma_h^2}{N_0}, \quad (6.98)$$

where  $\sigma_h^2$  is the energy in the received pulse,

$$\sigma_h^2 = \int_{-\infty}^{\infty} h^2(t) dt. \quad (6.99)$$

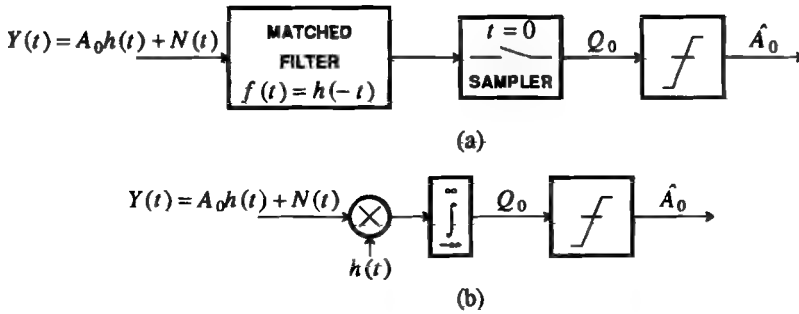
In the sequel, we choose  $K = 1$ , since any other choice affects the signal and noise terms equally.

To recap, the signal-to-noise ratio is maximized by choosing the receive filter to be (within a constant) the time-reversal of the received pulse shape  $f(t) = h(-t)$ . This filter, which has transfer function  $H^*(j\omega)$ , is called a matched filter. The matched filter performs perfect phase equalization, since the transfer function of the pulse at the output of the matched filter,  $|H(j\omega)|^2$ , is real-valued. It will be shown in Chapter 8 that under certain assumptions, for the isolated pulse case, this choice of receive filter minimizes the probability of error.

From (6.91), the matched-filter receiver output is the correlation of the received signal with the pulse  $h(t)$ ,

$$Q_0 = \int_{-\infty}^{\infty} Y(\tau) h(\tau) d\tau. \quad (6.100)$$

This implementation is known as the *correlation receiver*, and is shown in Figure 6-33 together with the equivalent matched-filter receiver. Viewing this in signal space (Section 2.6), we obtain an intuitive justification of the matched-filter receiver. The



**Figure 6-33.** Two equivalent baseband PAM receiver structures, (a) a matched-filter receiver and (b) a correlation receiver. For an isolated pulse, these receivers maximize the SNR at the slicer input.

receiver is taking the signal-space inner product of the received signal with the known pulse, or equivalently calculating the component of the received signal in the direction of the known pulse. Components in other directions in signal space must be due to the noise.

If  $h(t)$  is causal, as will usually be the case, then the matched filter is anti-causal. To implement it in practice,  $h(t)$  is assumed to be finite in length,

$$h(t) = 0 \text{ for } t \geq C \quad (6.101)$$

for some constant  $C$ , and the causal matched filter

$$f'(t) = h(C - t) \quad (6.102)$$

is implemented. A similar assumption is required to be able to compute the integral in the correlation receiver in finite time.

It should be emphasized that the preceding optimization ignores the effect of ISI. In general, if we use a matched filter as our receive filter, we will introduce ISI. However, no ISI occurs when the received pulse  $h(t)$  is confined to one symbol interval.

#### Exercise 6-4.

Show that if  $h(t) = 0$  for  $t < 0$  and  $T < t$ , then the pulse shape at the output of the matched filter,  $h(t) * h(-t)$ , is time-limited to two symbol intervals,  $-T \leq t \leq T$ , and furthermore goes to zero at  $t = -T$  and  $t = T$ . Thus, such a pulse shape at the output of the matched filter satisfies the Nyquist criterion.  $\square$

More generally, we can say that if the pulse shape at the output of the matched filter obeys the Nyquist criterion, then the matched filter is the optimal receive filter, in the sense that it maximizes the SNR. For a received pulse  $h(t)$ , the pulse at the output of the matched filter has Fourier transform  $|H(j\omega)|^2$ . The Nyquist criterion thus becomes, at the output of the matched filter,

$$S_h(j\omega) = \frac{1}{T} \sum_{m=-\infty}^{\infty} |H(j(\omega + m \cdot \frac{2\pi}{T}))|^2 = 1 \quad (6.103)$$

Of course, there will be no ISI if "1" is replaced by any constant, since that will simply scale the signal level at the slicer input. The quantity  $S_h(j\omega)$  is called the *folded spectrum* of the received pulse. It will play a key role in Chapters 7 through 10, where we consider ISI in detail. Equation (6.103) depends only on the magnitude  $|H(j\omega)|$ , as illustrated by the following example.

#### Example 6-33.

The raised cosine pulses given in (6.24) have a Fourier transform (6.25) that is real-valued and non-negative for all  $\omega$ . Therefore, a simple way to satisfy (6.103) is to use a pulse  $h(t)$  and receive filter  $f(t)$  with Fourier transforms equal to the square root of the raised cosine,

$$H(j\omega) = F(j\omega) = \sqrt{P(j\omega)}, \quad (6.104)$$

where  $P(j\omega)$  is given by (6.25). The corresponding time domain pulse shapes are [13],

$$h(t) = f(t) = \frac{4\alpha}{\pi\sqrt{T}} \frac{\cos((1+\alpha)\pi t/T) + T \sin((1-\alpha)\pi t/T)/(4\alpha t)}{1 - (4\alpha t/T)^2} \quad (6.105)$$

Convoluting such a pulse with itself will yield the raised cosine pulse of (6.24), so using such a pulse and receive filter results in no ISI at the receive filter output. Such pulses are called *square-root raised cosine* pulses. They are particularly easy to implement if the channel is assumed to be flat, so that the transmit pulse  $g(t)$  is the same as the received pulse  $h(t)$ .  $\square$

For the pulses in Exercise 6-4 and Example 6-33 the receive filter that maximizes the SNR also yields a pulse at its output that satisfies the Nyquist criterion. An extension to other cases will have to await Chapter 10.

### 6.6.2. Passband Case

In the passband PAM case, a received isolated pulse is

$$Y(t) = \sqrt{2} \operatorname{Re} \{ A_0 h(t) e^{j\omega_c t} \} + N(t), \quad (6.106)$$

where now the received pulse  $h(t)$  may be complex-valued (if the channel introduces dispersion) and the data symbol  $A_0$  is certainly complex-valued. The matched-filter or correlation receiver for this case, shown in Figure 6-34, is similar to Figure 6-33. We will now show that for an isolated pulse, these receivers maximize the SNR.

Considering again a receive filter  $f(t)$ , we showed in the development leading up to Figure 6-21 that the slicer input sampled at time zero is

$$Q_0 = A_0 \int_{-\infty}^{\infty} h(\tau) f(-\tau) d\tau + Z_0 \quad (6.107)$$

where  $Z_0$  is complex noise. Using reasoning similar to that leading to (6.56), it can be shown that this noise has variance

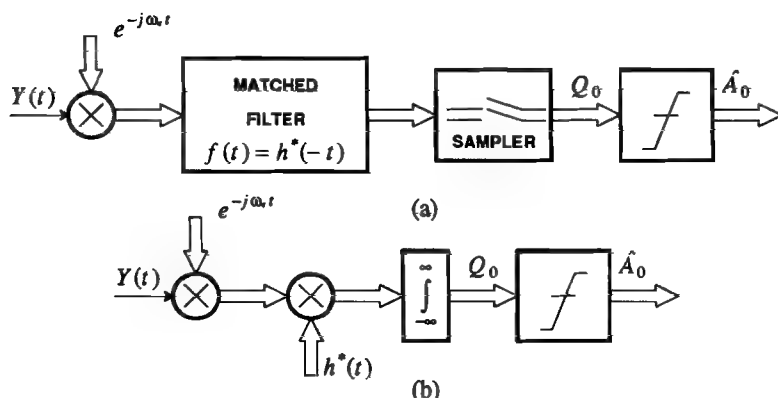


Figure 6-34. The (a) matched-filter and (b) correlation receiver shown for passband signals.

$$\sigma^2 = 2N_0 \cdot \int_{-\infty}^{\infty} |f(-\tau)|^2 d\tau \quad (6.108)$$

The SNR is given by

$$\text{SNR}_0 = \frac{\sigma_A^2 \left| \int_{-\infty}^{\infty} h(\tau) f(-\tau) d\tau \right|^2}{2N_0 \cdot \int_{-\infty}^{\infty} |f(-\tau)|^2 d\tau} \quad (6.109)$$

Again applying the Schwarz inequality (6.96), we conclude that

$$\text{SNR}_0 \leq \frac{\sigma_A^2 \sigma_h^2}{2N_0}, \quad (6.110)$$

where  $\sigma_h^2$  is given by (6.99). Now equality holds if and only if  $f(t) = h^*(-t)$ . Again, this filter is called a matched filter. For a baseband equivalent pulse  $h(t)$ , the baseband equivalent matched filter has impulse response  $h^*(-t)$  and transfer function (as in the baseband case)  $H^*(j\omega)$ . As in the baseband case, the Nyquist criterion at the matched filter output is satisfied if the folded spectrum  $S_h(j\omega)$  of (6.103) is a constant. The matched filter output sample at  $t = 0$  is equivalent to the cross-correlation of the complex baseband received signal with the waveform  $h^*(t)$ .

## 6.7. SPREAD SPECTRUM

*Spread spectrum* systems are PAM systems that deliberately use pulses with much more than the minimum bandwidth  $\pi/T$  required by the Nyquist criterion. From (6.98) and (6.110), the SNR achieved with a matched-filter or correlation receiver depends on the energy  $\sigma_h^2$  in the received pulse  $h(t)$ , but not on its bandwidth. So



from the perspective of SNR, there is no harm in using a pulse with a broad bandwidth, as long as a matched filter receiver is used. There are several reasons for using large bandwidth:

- Pulses with a broader spectrum are less sensitive to channel impairments that are highly localized in frequency. Such impairments arise, for example, with frequency-selective multipath fading.
- Spread spectrum signals are less vulnerable to *jamming*, in which a hostile party is trying to deliberately disrupt the communication.
- Spread spectrum signals can be concealed. By using very wide bandwidth pulses, these signals can be placed in regions of the spectrum already occupied by other signals, and in effect be masked by the other signals.
- Many spread spectrum users can share a common bandwidth without interfering much with one another.

Consider the jammer situation. Suppose that the bandwidth of the received pulse  $h(t)$  is  $B$ . Suppose that the total power of the jammer is limited to  $P_J$ , and that it transmits bandlimited white noise with power spectrum  $N_0 = P_J/2B$  within the bandwidth of the pulse. With a matched-filter receiver,

$$SNR_0 = \frac{\sigma_A^2 \sigma_h^2}{N_0} = 2B \frac{\sigma_A^2 \sigma_h^2}{P_J}. \quad (6.111)$$

As the bandwidth  $B$  increases, so does the SNR!

Recall that in the signal space view, the matched-filter or correlation receiver calculates the component of the received signal in the direction of the known pulse. The intuition behind spread spectrum is that it minimizes the effect of a particular impairment as long as that impairment has most of its energy in other directions in signal space. We will study this approach in more detail in Chapter 8.

## 6.8. ORTHOGONAL MULTIPULSE MODULATION

In baseband PAM, symbols  $A_k$  are multiplied by a pulse  $g(t)$  and combined for transmission,

$$S(t) = \sum_{k=-\infty}^{\infty} A_k g(t - kT). \quad (6.112)$$

A single pulse shape  $g(t)$  is used in one symbol interval, and amplitude modulated by the (possibly complex-valued) data symbol  $A_k$ . We can generalize this model by allowing the pulse shape in any symbol interval to be chosen from a set of  $N$  possibilities,  $\{g_n(t); 0 \leq n < N-1\}$ , to represent  $\log_2 N$  bits of information. The transmitted signal can then be written as

$$S(t) = \sum_{k=-\infty}^{\infty} g_{A_k}(t - kT) \quad (6.113)$$

where  $A_k$  takes on values in the set  $[0, N-1]$ . The data symbol thus indexes *which pulse* is transmitted in the  $k$ -th symbol interval, rather than the *amplitude* of the pulse that is transmitted. If the pulse set is *orthogonal* and *equal energy*, meaning that

$$\int_{-\infty}^{\infty} g_i(t) g_j^*(t) dt = \sigma_g^2 \delta_{i-j}, \quad (6.114)$$

for some constant  $\sigma_g^2$ , then we call this *orthogonal multipulse modulation*. In this section, we will initially follow the simplification of Section 6.6, and ignore the effects of ISI. Thus, we will transmit and receive a single isolated pulse, and design detection strategies that do not take into account the effects of ISI. After establishing some basic receiver structures for the isolated pulse, we will then generalize the Nyquist criterion to design orthogonal signals that avoid ISI.

For reasons seen shortly, orthogonal multipulse signaling has poor spectral efficiency, and hence is rarely used when bandwidth is at a premium. Nonetheless, it is valuable as a starting point for more elaborate techniques that combine it with PAM (Section 6.9).

### 6.8.1. Baseband Equivalent Model

For passband systems, we will often allow the pulses  $\{g_n(t); 0 \leq n \leq N-1\}$  to be complex baseband equivalents. In that case, the transmitted passband signal will be

$$X(t) = \sqrt{2} \operatorname{Re}\{e^{j\omega_c t} S(t)\}. \quad (6.115)$$

An alternative viewpoint is to define the passband equivalent pulses

$$\hat{g}_n(t) = \sqrt{2} \operatorname{Re}\{e^{j\omega_c t} g_n(t)\}. \quad (6.116)$$

These can then be used to form directly the passband signal

$$X(t) = \sum_{k=-\infty}^{\infty} \hat{g}_{A_k}(t - kT) \quad (6.117)$$

Both interpretations will be useful.

#### Exercise 6-5.

Show that if two complex-baseband waveforms are orthogonal as in (6.114), then their passband-equivalent real-valued waveforms are also orthogonal. Thus, orthogonality in baseband and passband are equivalent. You will need to assume that the carrier frequency is at least equal to the bandwidth of the baseband signal.  $\square$

### 6.8.2. The Correlation Receiver

We have assumed a simple receiver structure for PAM in which a receive filter eliminates out-of-band noise. A matched filter, with impulse response equal to the conjugate of the time-reversed pulse, was found to be the one receive filter that maximized SNR for the isolated pulse case. The matched-filter receiver can also be

interpreted as a correlation receiver, which cross-correlates the received signal with the transmitted pulse and feeds the resulting correlation to a slicer. In this section, we adapt this receiver structure to orthogonal multipulse. A receiver for the signal in (6.113) needs to distinguish different pulse *shapes* in each symbol interval, not just different pulse *amplitudes*.

Intuitively, the received signal can be cross-correlated with each candidate pulse shape. The pulse shape that correlates best with the signal can reasonably be assumed to be the one that was transmitted. Fortunately, the resulting receiver is equivalent to the matched-filter receiver. Thus, we have the happy situation that we can maximize SNR and distinguish orthogonal pulses *simultaneously*, at least for the isolated pulse case.

We will begin by considering real-valued pulses, which might be passband equivalent pulses as in (6.116). Let a received pulse be  $h_n(t)$  for some  $0 \leq n \leq N-1$ . We will assume the effect of the channel transfer function is benign, so that the received pulses  $h_n(t)$  are orthogonal and equal energy, or

$$\int_{-\infty}^{\infty} h_i(t) h_j^*(t) dt = \sigma_h^2 \delta_{i-j} . \quad (6.118)$$

For practical applications, such as multicarrier and code-division multiple access, discussed below, this assumption is usually valid. It is also obviously valid for channels with inherently flat frequency responses, such as certain radio channels.

As in Section 6.6, ignore ISI by considering a received isolated pulse. Assume the received signal  $Y(t)$  is corrupted by Gaussian white noise,

$$Y(t) = h_n(t) + N(t). \quad (6.119)$$

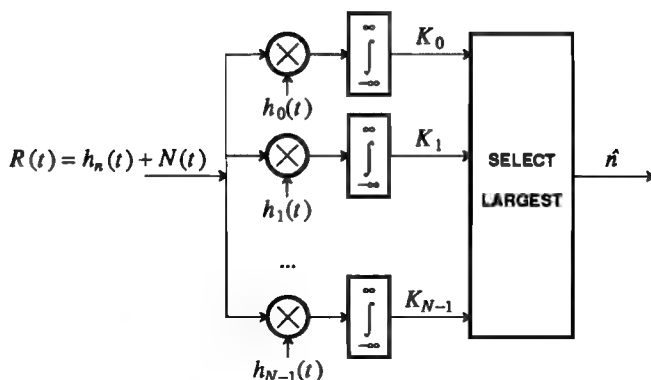
The correlation receiver forms  $N$  cross-correlations,

$$K_i = \int_{-\infty}^{\infty} Y(t) h_i(t) dt = \int_{-\infty}^{\infty} h_n(t) h_i(t) dt + \int_{-\infty}^{\infty} N(t) h_i(t) dt , \quad (6.120)$$

for  $0 \leq i \leq N-1$ . Since the possible received pulses  $h_i(t)$  are orthogonal, and pulse  $n$  was transmitted,  $K_n$  will be equal to  $\sigma_h^2$  plus noise, while  $K_i$  for  $i \neq n$  will be noise only. So it makes intuitive sense to choose the maximum  $K_i$  to decide which pulse was transmitted.

This correlation receiver is illustrated in Figure 6-35. It works very well for orthogonal multipulse since, by the orthogonality property, the output of a cross-correlation against one pulse shape will have a zero signal component if any of the other pulse shapes is actually transmitted. From a signal-space perspective, each  $K_i$  looks only in the direction of  $h_i(t)$  in signal space by forming an inner product (cross-correlation) of  $h_i(t)$  with the received signal. Our intuition will be confirmed in Chapter 9, where the correlation receiver is shown to be optimal under the assumption of additive white Gaussian channel noise.

For the passband case, we should use the passband-equivalent pulse  $\hat{h}_n(t) = \text{Re}\{ e^{j\omega_c t} h_i(t) \}$  in place of  $h_n(t)$  in (6.120), to obtain



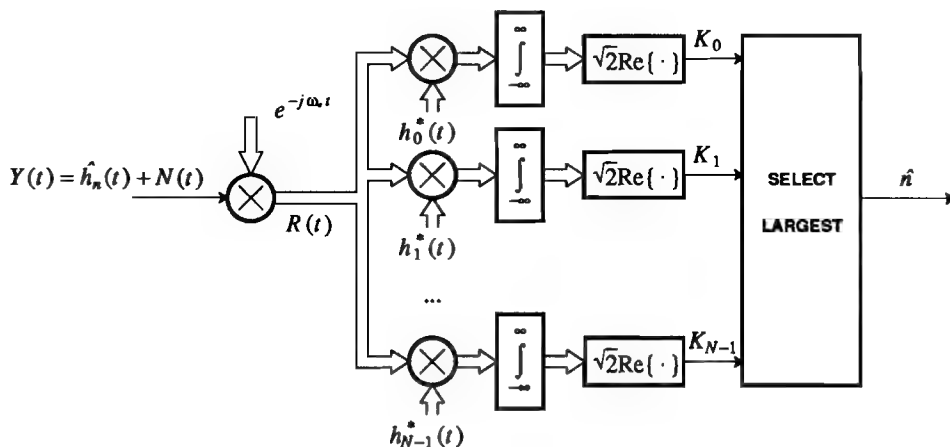
**Figure 6-35.** An isolated-pulse correlation receiver for baseband multipulse transmission, where the received pulses are assumed to be real and orthogonal.

$$\begin{aligned}
 K_i &= \sqrt{2} \int_{-\infty}^{\infty} Y(t) \operatorname{Re} \{ e^{j\omega_c t} h_i(t) \} dt = \sqrt{2} \int_{-\infty}^{\infty} Y(t) \operatorname{Re} \{ e^{-j\omega_c t} h_i^*(t) \} dt \\
 &= \sqrt{2} \operatorname{Re} \left\{ \int_{-\infty}^{\infty} R(t) h_i^*(t) dt \right\}
 \end{aligned} \tag{6.121}$$

where the demodulated received signal is

$$R(t) = Y(t) e^{-j\omega_c t}. \tag{6.122}$$

This interpretation of the receiver is shown in Figure 6-36. The receive signal is



**Figure 6-36.** An isolated pulse correlation receiver for passband orthogonal multipulse, using baseband equivalent pulses.

demodulated with a complex exponential and correlated with the baseband equivalent pulse. The scaled real part of the result is used to make the decision. This receiver structure works entirely with the baseband equivalent pulses.

The correlation receiver can be implemented as a set of matched filters. Define

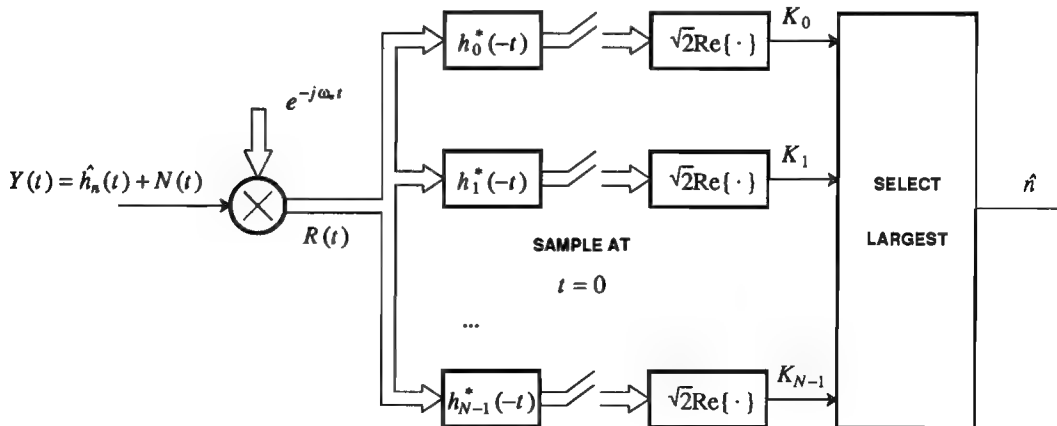
$$f_i(t) = h_i^*(-t) \quad (6.123)$$

and note that

$$\begin{aligned} K_i &= \sqrt{2} \operatorname{Re} \left\{ \left[ f_i(t) * R(t) \right]_{t=0} \right\} = \sqrt{2} \operatorname{Re} \left\{ \int_{-\infty}^{\infty} R(\tau) f_i(-\tau) d\tau \right\} \\ &= \sqrt{2} \operatorname{Re} \left\{ \int_{-\infty}^{\infty} R(\tau) h_i^*(\tau) d\tau \right\}. \end{aligned} \quad (6.124)$$

Hence the correlations  $K_i$  can be computed by sampling the output of a filter with impulse response equal to the time-reversed conjugated pulse. A matched-filter receiver is shown in Figure 6-37.

Detection of an isolated pulse is only the beginning, of course. To detect a sequence of pulses, the matched-filter receiver in Figure 6-37 can be modified so that samples are taken at multiples of  $T$ , rather than just once at  $t = 0$ . As we will see in Chapter 9, this will prove to be optimal if such sampling does not result in intersymbol interference.



**Figure 6-37.** A matched-filter receiver for an isolated pulse in multipulse transmission, using baseband-equivalent pulses.

### 6.8.3. The Generalized Nyquist Criterion

There is a fundamental lower bound on the bandwidth required by an orthogonal multipulse signal, assuming that we wish to avoid ISI. The Nyquist criterion, discussed in Section 6.2, states that for baseband PAM with symbol rate  $T$ , the minimum signal bandwidth is  $\pi/T$  radians/sec or  $1/2T$  Hz. We can now generalize the Nyquist criterion and show that the minimum bandwidth of orthogonal multipulse is  $N\pi/T$  radians/sec or  $N/2T$  Hz. Thus, the requirement that there be  $N$  orthogonal pulses in the symbol interval increases the minimum bandwidth requirement by  $N$ .

Assume the receiver structure of Figure 6-37, and for an isolated-pulse input, sample the matched-filter outputs at all integer multiples of  $T$ . To avoid ISI, if the signal input is pulse  $h_n(t)$ , then the samples at the output of the filter matched to  $h_n(t)$  must satisfy the ordinary Nyquist criterion,

$$h_n(t) * h_n^*(-t) \Big|_{t=kT} = \delta_k, \quad 0 \leq n \leq N-1. \quad (6.125)$$

In addition, to avoid crosstalk between pulses, if  $h_n(t)$  is the input to a filter matched to pulse  $h_l(t)$ , for  $l \neq n$ , then the output sampled at  $t = kT$  must be zero for all  $k$ ,

$$h_n(t) * h_l^*(-t) \Big|_{t=kT} = 0, \quad l \neq n, \quad -\infty < k < \infty. \quad (6.126)$$

These conditions can be written together in a compact form,

$$h_n(t) * h_l^*(-t) \Big|_{t=kT} = \delta_k \delta_{l-n}. \quad (6.127)$$

We can express these conditions in terms of an equivalent frequency-domain criterion. Let  $h_n(t)$  have Fourier transform  $H_n(j\omega)$ . When we input  $h_n(t)$  to a filter matched to  $h_l(t)$ , the output has Fourier transform  $H_n(j\omega)H_l^*(j\omega)$ . Sample this at  $t = kT$ , the discrete-time Fourier transform has to be unity for  $l = n$  and zero for  $l \neq n$ , and hence

$$\frac{1}{T} \sum_{m=-\infty}^{\infty} H_n(j(\omega + m\frac{2\pi}{T})) H_l^*(j(\omega + m\frac{2\pi}{T})) = \delta_{l-n}. \quad (6.128)$$

Equation (6.128) is called the *generalized Nyquist criterion*. Using this, we can show that in order to avoid ISI, the aggregate of  $N$  orthogonal pulses occupies a minimum bandwidth of  $N\pi/T$ , or  $N$  times the minimum bandwidth of ordinary PAM.

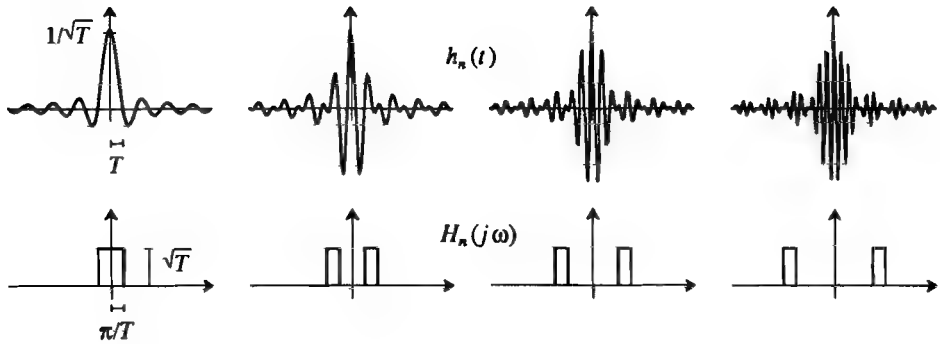
First, we show that a bandwidth of  $N\pi/T$  is sufficient to satisfy (6.128) by displaying a pulse set that meets the criterion.

#### Exercise 6-6.

Let

$$h_n(t) = \frac{1}{\sqrt{T}} \left[ \frac{\sin(\pi t/2T)}{\pi t/2T} \right] \cos \left[ (n + 1/2) \frac{\pi}{T} t \right] \quad (6.129)$$

for  $n = 0, \dots, N-1$ . Show that these pulses are ideally bandlimited to the range  $\pi n/T \leq |\omega| < \pi(n+1)/T$ , as shown in Figure 6-38, so that the aggregate bandwidth occupied by the first  $N$  pulses is  $N\pi/T$ . Also show that they satisfy (6.128).  $\square$



**Figure 6-38.** Time domain (top) and frequency domain (bottom) plots of the pulses in (6.129) for  $n = 0, 1, 2$ , and  $3$ .

Since these are ideally bandlimited pulses, they are not practical. Nonetheless, they demonstrate that a bandwidth of  $N\pi/T$  is sufficient to satisfy (6.128). In Appendix 6-B, we also show that this bandwidth is necessary.

### Spectral Efficiency of Orthogonal Multipulse

The minimum bandwidth of orthogonal multipulse with no ISI is  $N\pi/T$  radians/sec or  $N/2T$  Hz. Consequently the best spectral efficiency is

$$\nu = \frac{\log_2(N)}{T \cdot (N/2T)} = 2 \frac{\log_2(N)}{N} \quad (6.130)$$

using (6.8). The maximum spectral efficiency, easily established by differentiating  $\nu$  with respect to  $N$ , is at  $N = e$ , and the resulting spectral efficiency is  $\nu = 2/(e \cdot \log_e 2) = 1.07$ . Of course,  $N$  must be an integer, so the best we can do is  $N = 3$ , where  $\nu = 1.056$  bits/sec-Hz.  $N = 2$  and  $N = 4$ , both with  $\nu = 1$  bits/sec-Hz, are almost as good. As we increase from  $N = 4$ , the spectral efficiency decreases because the  $\log_2(N)$  increases more slowly than  $N$ .

PAM can be much more spectrally efficient, assuming that the signal-to-noise ratio allows us to increase the number of bits/symbol, because that increase in bits/symbol comes without any impact on either the bandwidth or the symbol interval. On the other hand, orthogonal signaling is inherently less susceptible to noise, because we are in effect doing a binary rather than  $M$ -ary amplitude modulation of the transmitted pulse. Any complete comparison of modulation techniques must take into account both spectral efficiency and noise immunity; this will be undertaken in Chapter 8.

## A Set of Orthonormal Pulses

The ideally bandlimited pulse set in Figure 6-38 is not realizable. Practical pulse sets designed by Chang [14], achieve close to the minimum bandwidth promised by the generalized Nyquist criterion. We will derive these pulses, relegating many details to Appendix 6-B.

Let  $q(t)$  be a pulse shape chosen such that  $q(t) * q(-t)$  (the pulse shape at the output of a matched filter) satisfies the Nyquist criterion for symbol rate  $1/2T$ ; that is,

$$\int_{-\infty}^{\infty} q(t)q(t - 2kT) dt = \delta_k. \quad (6.131)$$

The minimum bandwidth of  $q(t)$  is  $\pi/2T$  radians/sec, but to allow a gradual rolloff, assume that it has twice the minimum bandwidth, or  $\pi/T$ . An example of a  $|Q(j\omega)|^2$  satisfying these conditions is shown in Figure 6-39a. Note that only the magnitude of  $Q(j\omega)$  is constrained by this condition, not the phase. We will choose the phase later to satisfy the generalized Nyquist criterion.

For  $1 \leq n \leq N$ , define a set of pulse shapes

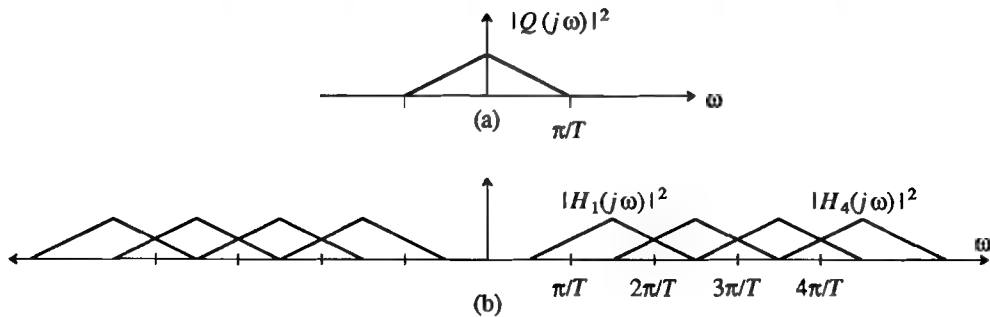
$$h_n(t) = q(t) \cdot \cos[(n + 1/2)\pi t/T]. \quad (6.132)$$

This is a generalization of (6.129).

### Exercise 6-7.

Show that as long as  $q(t)$  satisfies (6.131), then for each  $n$ , the matched filter output  $h_n(t) * h_n(-t)$  satisfies the ordinary Nyquist criterion (6.125).  $\square$

To show that the  $\{h_n(t)\}$  satisfy the generalized Nyquist criterion, we must verify that (6.126) holds. This requires some of the same machinery used to prove the



**Figure 6-39.** (a) The Fourier transform of a pulse at the output of a matched filter that satisfies the Nyquist criterion for symbol rate  $1/2T$ . (b) The magnitude squared of the Fourier transforms of a set of four orthonormal pulses that satisfy the generalized Nyquist criterion at symbol rate  $T$ , but unlike the pulses in Figure 6-38 overlap in the frequency domain.



minimum bandwidth of orthogonal multipulse, so we defer the details to Appendix 6-B. In particular, in the appendix we show that (6.126) holds if  $Q(j\omega)$ , the Fourier transform of  $q(t)$ , satisfies

$$\operatorname{Re}\{Q(j\omega)Q(j(\pi/T - \omega))\} = 0, \quad 0 \leq \omega \leq \pi/2T. \quad (6.133)$$

This can be satisfied by adjusting the phase of  $Q(j\omega)$  to have a particular symmetry about  $\pi/2T$ , without regard to the magnitude. The magnitude can be chosen to satisfy (6.131) and simultaneously the phase can be chosen to satisfy (6.133). To see this, write  $Q(j\omega)$  in terms of its magnitude and phase,  $Q(j\omega) = A(\omega)e^{j\theta(\omega)}$ . Then

$$\begin{aligned} \operatorname{Re}\{Q(j\omega)Q(j(\pi/T - \omega))\} \\ = A(\omega)A(\pi/T - \omega)\cos(\theta(\omega) + \theta(\pi/T - \omega)). \end{aligned} \quad (6.134)$$

This function will be zero for all  $0 \leq \omega \leq \pi/2T$  if

$$\theta(\omega) + \theta(\pi/T - \omega) = \pm \pi/2 \quad (6.135)$$

over the same range of  $\omega$ .

#### Example 6-34.

As an example of a phase function satisfying this constraint, let

$$\theta(\omega) = -\omega T/2 + \gamma(\omega) \quad (6.136)$$

where  $\gamma(\omega)$  is any function that has odd symmetry about  $\pi/2T$ . Then,

$$\theta(\omega) + \theta(\pi/T - \omega) = -\pi/2 + \gamma(\omega) + \gamma(\pi/T - \omega) = -\pi/2. \quad (6.137)$$

This phase function includes a linear-phase term, corresponding to a delay, plus another arbitrary phase term meeting the odd-symmetry constraint. There is thus considerable freedom in choosing this phase function.  $\square$

To summarize, we have defined a class of pulse sets (6.132) that satisfy the generalized Nyquist criterion. The magnitude  $|Q(j\omega)|$  has been chosen to force  $q(t) * q(-t)$  to satisfy the ordinary Nyquist criterion at symbol rate  $1/2T$ , and the phase of  $Q(j\omega)$  has been chosen to force the pulses to be orthogonal as required.

The set of pulses  $\{h_n(t); 1 \leq n \leq N\}$  cover the frequency interval  $[\pi/2T, (N + 3/2)\pi/T]$ , for a total aggregate bandwidth of  $(N + 1)\pi/T$ . This is only slightly larger than the minimum bandwidth  $N\pi/T$ , for large  $N$ . This increase in bandwidth is the small price paid for achieving gradual rolloff.

### 6.8.4. Frequency-Shift Keying

Minimum bandwidth orthogonal multipulse pulses like those given in (6.129) are not practical because the pulses are ideally bandlimited, as shown in Figure 6-38. The generalization given in (6.132) allows for more gradual rolloff, although the bandwidth of the pulses is still strictly limited. The signaling schemes simplest to implement result from further relaxing the bandwidth constraint. For example, *frequency shift keying (FSK)* is a form of orthogonal multipulse modulation where the pulses are not bandlimited at all. It is often used where hardware simplicity of the receiver is important, the channel has a high degree of nonlinear distortion, or it is not

desired or possible to generate an accurate carrier replica. FSK will be inferior to PAM in both spectral efficiency and noise immunity, but these are not always the overriding considerations.

### Example 6-35.

In *binary FSK*, two distinct carrier frequencies are used. An example is shown in Figure 6-40, in which a higher frequency is used to represent a binary "1" and a lower frequency is used to represent a binary "0". This modulation uses the two pulse shapes shown below:



It is easy to show that these two pulse shapes are orthogonal if each contains an integral number of cycles in a symbol interval.  $\square$

Because FSK is a special case of orthogonal multipulse modulation, we can use the correlation (or the equivalent matched filter) receiver.

### Example 6-36.

The binary FSK pulses of the previous example, properly normalized, can be written

$$\begin{aligned} g_0(t) &= \sqrt{2/T} \sin(\omega_0 t) w(t) \\ g_1(t) &= \sqrt{2/T} \sin(\omega_1 t) w(t), \end{aligned} \quad (6.138)$$

where

$$w(t) = \begin{cases} 1; & 0 \leq t < T \\ 0; & \text{otherwise} \end{cases} \quad (6.139)$$

is a rectangular window function. Assume that the channel is benign, so that the received pulses  $h_i(t)$  are equal to the transmitted pulses  $g_i(t)$  for  $i = 0, 1$ . The matched filters,  $f_i(t) = h_i(-t) = g_i(-t)$ , have finite impulse response, but are non-causal. We can define causal matched filters,

$$\begin{aligned} f'_0(t) &= f_0(t - T) = g_0(T - t) \\ f'_1(t) &= f_1(t - T) = g_1(T - t). \end{aligned} \quad (6.140)$$

For an isolated pulse, the sampling is delayed to  $t = T$  instead of  $t = 0$ .  $\square$

More generally, a set of  $M$  frequencies can be chosen, and the resulting pulse shapes will be orthogonal if each frequency is chosen with an integral number of cycles per



Figure 6-40. An example of a binary FSK signal.

symbol interval (this condition is sufficient, but not necessary). However, because of the inherent spectral inefficiency of orthogonal multipulse modulation, it is rare to use more than two frequencies (which has approximately the best spectral efficiency, as we saw earlier).

The primary advantages of FSK are in areas other than spectral efficiency:

- *Incoherence.* For detection of passband PAM signals we have thus far assumed that the exact carrier frequency and phase is available at the receiver to perform the demodulation. However, as we will show in Chapter 16, carrier recovery is far from trivial, especially on certain channels where the received carrier phase varies rapidly. Examples of such channels are optical fiber (see Section 6.10) and radio links to rapidly moving vehicles such as aircraft. Effective FSK receivers can be designed that make no attempt to recover the carrier phase. Such receivers are said to be *incoherent*.
- *Ease of implementation.* Very simple FSK modems are possible mainly because incoherent detection is feasible. In addition, it is also sometimes possible to avoid timing recovery.
- *Immunity from certain nonlinearities.* Most FSK modulation techniques result in a *constant envelope*, in which information is carried by the zero crossings of the signal alone. Some channels, such as those using RF amplifiers operating at or near saturation, hard-limit the signal.

**Example 6-37.**

The CCITT V.21 300 b/s voiceband modem standard uses binary FSK with frequencies of  $1080 \pm 100$  Hz for transmission in one direction and  $1750 \pm 100$  Hz for transmission in the other direction. The motivation for using FSK in this case is the hardware simplicity of the receiver and the fact that no carrier or timing recovery is required. This standard was developed in an age of more expensive hardware built with discrete devices. □

**Example 6-38.**

The transmit power of a satellite is limited, and more power can be generated if the transponder is operated in or near its nonlinear saturation region. The transponder is essentially a peak-power limiter, and a *constant envelope* modulation method such as FSK can achieve a higher average power for a given peak power. □

FSK also has significant disadvantages, other than spectral inefficiency:

- *3dB penalty.* Binary FSK suffers a 3 dB penalty in SNR required to achieve a given probability of error relative to binary antipodal PAM, as will be shown in Chapter 8.
- *Difficult equalization.* The basic FSK signal is not usually a linear function of the data, so existing adaptive equalization techniques (Chapter 11) fail. Compensation for channel distortion is therefore more difficult than for PAM signals. FSK is therefore primarily limited to channels such as fiber optics and satellite where frequency dispersion and selective fading are not a problem.

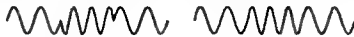
- *Difficult analysis.* Also because it is nonlinear, FSK is difficult to analyze. For example, the spectrum is difficult to determine analytically, as is the probability of error for commonly-used suboptimal receiver structures.

### FSK Receivers

The correlation receiver for orthogonal multipulse modulation is optimal for FSK in white Gaussian noise, as will be shown in Chapter 9. However, one of the major advantages of FSK is the simple suboptimal receivers that are possible. For example, the receiver shown in Figure 6-41 discriminates between transmitted frequencies simply by measuring the density of zero crossings. The performance of this receiver is difficult to analyze, partly because Gaussian noise on the channel does not imply Gaussian noise at the input to the slicer. (A similar receiver is successfully analyzed in [15].) Fortunately, the correlation receiver is easier to analyze, and provides a lower bound on the probability of error for any suboptimal receiver. This will be developed further in Chapter 8.

### Continuous-Phase FSK

The FSK transmitted signal can have either discontinuous phase or continuous phase, depending on the signal design, as illustrated below:



*Continuous phase*, shown on the right, is desirable, since the high frequency components are reduced. This is important for a bandlimited channel, and particularly important when the channel is nonlinear.

#### Example 6-39.

In a satellite application, we would like to drive the transmitter RF amplifier hard enough into saturation that significant nonlinearity results. There are two alternatives for the ordering of the nonlinearity (which we model as memoryless) and bandpass filtering, as shown in Figure 6-42. In the first alternative, Figure 6-42a, the bandpass filtering is done after the nonlinearity. For this case, phase discontinuities and the nonlinearity have relatively little interaction. Unfortunately, this case is also often impractical, since the nonlinearity is introduced at high frequencies and therefore the bandpass filter would have to be introduced at the same frequencies. Such a bandpass filter to be effective would have an incredibly high  $Q$ . Furthermore, the bandpass filter would follow the power amplifier, which

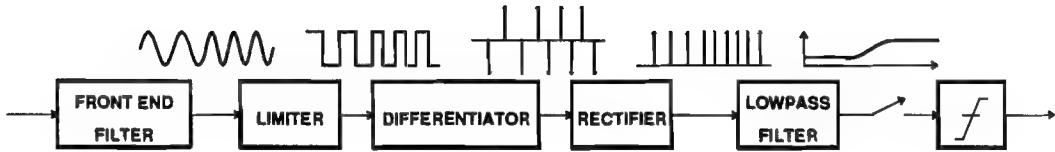
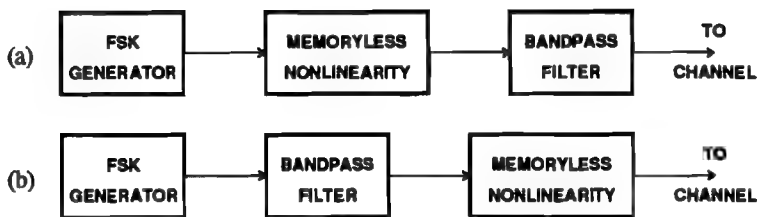


Figure 6-41. A zero-crossing detector for binary FSK with the accompanying waveforms.



**Figure 6-42.** Two alternatives for the ordering of bandpass filtering and nonlinearity in a microwave radio transmitter.

introduces the nonlinearity. This means it can only use passive components, and these components would have to be capable of dissipating considerable power. Also, the loading of the power amplifier could become a problem. The second case is shown in Example 6-39b. This is more practical, since in this case the bandpass filtering can be performed at lower frequencies, prior to the modulator. Now there is unfortunately considerable interaction between phase transitions and the nonlinearity, since the bandpass filter will convert the phase transitions into an amplitude transient, which might be quite large. The power of the signal at RF will then have to be backed off in order to keep the amplitude of these transients within the peak power limitation, thereby reducing the average signal power and increasing the error rate at the receiver. In this case continuous phase is preferred.  $\square$

For pulses of the form (6.138) to result in a continuous-phase FSK signal, each pulse must traverse an integer number of cycles, so

$$\frac{\omega_i T}{2\pi} = M_i \quad (6.141)$$

where  $M_i$  is an integer. For maximum spectral efficiency, it also helps to minimize the *frequency separation* between signaling frequencies  $\omega_i$ . We want minimum frequency separation while maintaining phase continuity and orthogonality of pulses.

#### Example 6-40.

For the binary FSK pulses of (6.138), continuous phase requires

$$\omega_0 T / 2\pi = M_0 \quad \text{and} \quad \omega_1 T / 2\pi = M_1 \quad (6.142)$$

where  $M_0$  and  $M_1$  are integers. Minimum frequency separation  $|\omega_0 - \omega_1|$  requires that  $|M_0 - M_1| = 1$ . One of the pulses must traverse exactly one more cycle than the other. Two such pulses are superimposed below:



The pulse frequencies satisfy

$$2\omega_d = |\omega_1 - \omega_0| = 2\pi/T \quad (6.143)$$

where  $\omega_d$  is called the *peak deviation* from the nominal carrier frequency  $\omega_c = (\omega_1 + \omega_0)/2$ . Such pulses are easily shown to be orthogonal (see Problem 6-19). A smaller frequency separation than that given by (6.143), maintaining both phase continuity and orthogonality,

is possible. It requires generalizing the FSK model, and is done below in Section 6.8.5.  $\square$

### Exercise 6-8.

As in Example 6-36, assume the channel is benign, so that the received pulses are equal to the transmitted pulses given in (6.138). Assume as in Example 6-40 that

$$\omega_0 = 2\pi M_0/T \quad \text{and} \quad \omega_1 = 2\pi M_1/T, \quad (6.144)$$

where  $M_0$  and  $M_1$  are arbitrary integers. Show that with these frequencies, the pulses (6.138) satisfy (6.127), and hence satisfy the generalized Nyquist criterion.  $\square$

For continuous-phase FSK signals with  $N$  pulses, Example 6-40 suggests that the frequencies of pulses must differ by integer multiples of  $2\pi/T$ . If the pulses have frequencies  $\omega_0 < \omega_1 < \dots < \omega_{N-1}$ , then

$$\omega_i - \omega_{i-1} = 2\pi/T; \quad \text{for } 1 \leq i \leq N-1 \quad (6.145)$$

is the minimum separation. This frequency spacing of  $2\pi/T$  between pulses is twice as large as that achieved by the ideally bandlimited multipulse pulses of (6.129) and the Chang pulses of (6.132).

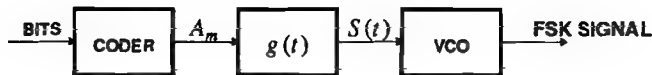
### Example 6-41.

For binary FSK, a frequency spacing of  $\pi/T$ , equivalent to the spacing of the Chang pulses in (6.132), is possible (see Problem 6-19). However, when more than two pulses of the form (6.138) are used, they will not in general be orthogonal with such spacing.  $\square$

A continuous-phase FSK transmitter can be implemented using a *voltage-controlled oscillator* (VCO, Chapter 15) driven by a baseband PAM signal, as shown in Figure 6-43. The VCO frequency varies about a nominal carrier  $\omega_c$  and automatically maintains phase continuity. The output signal can be written

$$X(t) = K \cos \left[ \omega_c t + \omega_d \int_{-\infty}^t S(\tau) d\tau \right]. \quad (6.146)$$

A signal of this form is called *continuous-phase modulation* (CPM). The baseband data signal is written



**Figure 6-43.** An implementation of an FSK transmitter using a voltage controlled oscillator (VCO).

$$S(t) = \sum_{k=-\infty}^{\infty} A_k g(t - kT). \quad (6.147)$$

The data signal  $S(t)$  is usually normalized to peak at unity,  $|S(t)| \leq 1$ , in which case  $\omega_d$  is the peak frequency deviation. When  $g(t)$  is rectangular, CPM is called *continuous-phase FSK* (CPFSK). Other pulse shapes are also used and can lead to significantly improved performance.

The nonlinear relationship between the data signal  $S(t)$  and the FSK signal is clearly seen in (6.146). Among other effects, this nonlinear relationship makes it very difficult to find an expression for the power spectrum of a general CPFSK signal. Suitable derivations and plots are given elsewhere (see for example [16,15]). The bandwidth of an FSK signal is usually wider than a passband PAM signal with the same symbol rate, except when *minimum shift keying* (MSK) is used. MSK, discussed in the following subsection, is characterized by a frequency deviation  $\omega_d$  that is half that predicted by (6.145), equivalent to the spacing of the Chang pulses (6.132).

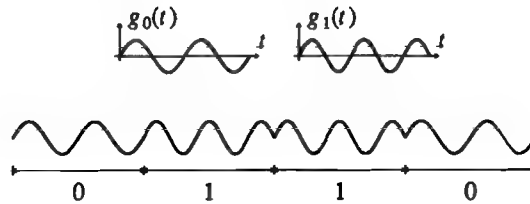
### 6.8.5. Minimum Shift Keying

In the previous section we determined that orthogonal FSK pulses have continuous phase if (6.145) is satisfied. However, it is possible to reduce the frequency separation still further while maintaining orthogonality and phase continuity using the model in (6.146). This reduces the bandwidth occupied by the main portion of the signal. When the frequency separation is half of that predicted by (6.145), the modulation technique is called *Minimum Shift Keying* (MSK).

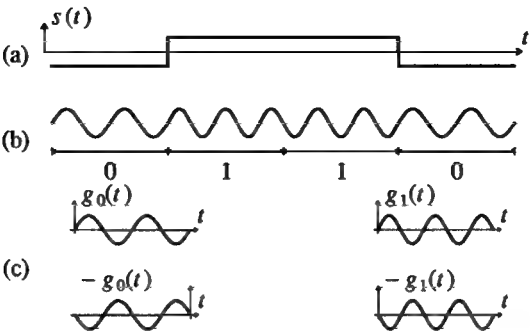
MSK was invented by Doelz and Heald [17], and has been used in some microwave radio systems. It is sometimes called *fast FSK* (FFSK) because the spectral efficiency is higher than more traditional FSK signals. The separation in frequency between pulses is equivalent to that of the minimum-bandwidth pulses (6.129) and the Chang pulses (6.132). By introducing memory from one symbol to the next, it cleverly maintains phase continuity while decreasing the frequency separation. If the MSK pulses have frequencies  $\omega_0 < \omega_1 < \dots < \omega_{N-1}$ , then

$$\omega_i - \omega_{i-1} = \pi/T; \text{ for } 1 \leq i \leq N-1. \quad (6.148)$$

We will illustrate MSK for the binary case in the following example.



**Figure 6-44.** a. Two orthogonal FSK pulses with frequency separation  $|\omega_1 - \omega_0| = \pi/T$ . b. An FSK signal formed using the multipulse model. Note the discontinuous phase.

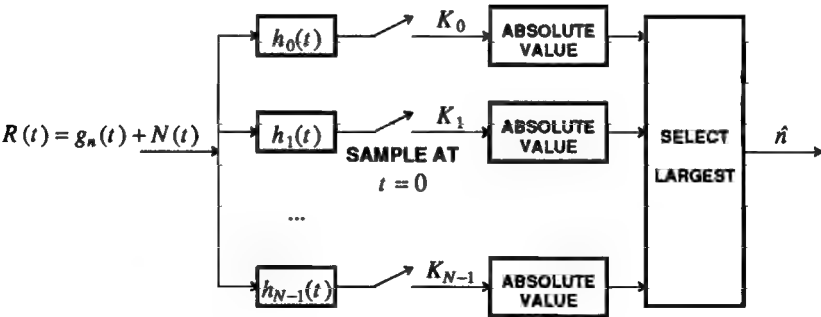


**Figure 6-45.** This figure shows the relevant signals for continuous-phase binary MSK. The peak frequency deviation is  $\omega_d = \pi/2T$ . a. A data signal. b. The corresponding MSK signal. c. The pulses. Notice that each bit value has two possible pulses which are negatives of one another.

**Example 6-42.**

Suppose that  $|\omega_1 - \omega_2| = \pi/T$  in the binary FSK example. The peak frequency deviation is  $\omega_d = \pi/2T$ . The higher frequency sinusoid traverses half a cycle more than the lower frequency sinusoid within one symbol interval. The multipulse model of (6.113) for FSK would use two pulses like those in Figure 6-44a, resulting in the signal shown in Figure 6-44b, which has discontinuous phase. Using the continuous-phase model of (6.146), however, the corresponding signals are shown in Figure 6-45. The frequency separation is half the minimum frequency separation of Example 6-40, yet the pulses are orthogonal (see Problem 6-20).  $\square$

The correlation or matched-filter receiver needs to be modified slightly to accommodate the MSK model. Notice from Figure 6-45 that



**Figure 6-46.** A matched-filter receiver for an MSK signal. However, we will see in Section 12.4 that this receiver is not optimal.



$$g_i(t) = \pm \sin(\omega_i t) w(t), \quad (6.149)$$

where  $w(t)$  is given by (6.139). There are two signals with opposite polarity for each symbol. Since the signals with opposite polarities bear the same information, the matched-filter receiver should compare the *absolute value* of the sampled outputs, as shown in Figure 6-46.

It should be emphasized at this point that MSK is not orthogonal multipulse. From (6.149), the two possible transmitted pulses are  $\sin(\omega_i t)$  and  $-\sin(\omega_i t)$ , which are not orthogonal. In fact (see Section 6.9 below), MSK is a generalization that combines PAM with orthogonal multipulse. We will see additional examples of combined modulation formats in Section 6.9.

The receiver in Figure 6-46 throws away potentially useful information, since based on the history of the received signal, we could infer which of the two polarities would be expected. Making use of this information can reduce the probability of error. In Section 12.4 we will show how to take advantage of this additional information.

The pulses in (6.149) can be re-written

$$\begin{aligned} g_i(t) &= \pm \sin(\omega_c t + b \frac{\pi t}{2T}) w(t) \\ &= \sin(\omega_c t + b \frac{\pi t}{2T} + \phi) w(t), \end{aligned} \quad (6.150)$$

where  $b = \pm 1$  determines the transmit frequency and  $\phi = 0$  or  $\pi$  depending on which phase is being transmitted. The nominal carrier frequency is

$$\omega_c = (\omega_0 + \omega_1)/2. \quad (6.151)$$

One representation for the transmitted MSK signal is therefore

$$X(t) = \sum_{k=-\infty}^{\infty} \sin(\omega_c t + b_k \frac{\pi t}{2T} + \phi_k) w(t - kT), \quad (6.152)$$

where  $b_k$  is determined by the data and  $\phi_k$  ensures phase continuity.

#### Exercise 6-9.

Show that to maintain phase continuity we need

$$\phi_k = \phi_{k-1} + (b_{k-1} - b_k) \pi k / 2 \mod 2\pi. \quad (6.153)$$

□

Expression (6.153) explicitly shows the dependence of the phase in each symbol interval on the data.

The matched-filter receiver of Figure 6-46 performs roughly as well with binary MSK as with FSK. This will be studied in Chapter 8. In Section 12.4, we will show that the performance can be improved to make MSK better than FSK, in that the probability of error will be lower and the bandwidth will be smaller. This is accomplished by using the information thrown away by the absolute values in Figure 6-46, the error probability of MSK becomes essentially equivalent to PSK.

Will the zero-crossing detector in Figure 6-41 perform as well with MSK as with other orthogonal FSK signals? From Figure 6-45, the higher-frequency signal of a binary MSK signal set has only one more zero crossing than the lower-frequency signal. A larger frequency deviation would increase the difference in the number of zero crossings, and hence, apparently, decrease the probability of error. It would appear, in fact, that we could reduce the probability of error arbitrarily by increasing the frequency separation! However, the larger frequency separation means that the signal has greater bandwidth, so with a simple bandpass receive filter, the noise admitted to the system is proportionally greater. It is possible to carefully design the front end filter in Figure 6-41 so that it does not admit more noise, but then the noise spectrum is no longer flat, and the noise will have oscillations that tend to make spurious zero crossings more likely for a given noise power [15].

An interesting property of MSK signals is that they can be interpreted as passband PAM signals where the quadrature component is delayed half a symbol interval with respect to the in-phase component. Such signals are called *offset keyed QAM* (OQAM or OK-QAM) or *offset keyed phase-shift keying* (OPSK or OK-PSK). MSK is a special case (see Problem 6-22). They can also be interpreted as combined PAM and multipulse (Section 6.9).

### 6.8.6. Incoherent Receivers

The matched-filter and correlation receivers require that the receiver accurately know the frequency and phase of the sinusoids used to form the FSK pulses. Such receivers are said to be *coherent*.

#### Exercise 6-10.

Consider the reception of a single isolated pulse,

$$g_0(t) = \sqrt{2/T} \sin(\omega_0 t) w(t), \quad (6.154)$$

where  $\omega_0 = M_0 2\pi/T$  for some integer  $M_0$ , and  $w(t)$  is a rectangular window over  $(0, T)$ , as in Example 6-36. Assume a benign channel so that  $h_0(t) = g_0(t)$  is received. Suppose that the receiver cannot accurately determine the phase of the sinusoid in (6.154), so it uses the erroneous matched filter

$$f(t) = \sqrt{2/T} \sin(-\omega_0 t + \theta) w(t), \quad (6.155)$$

where  $\theta$  is the phase error. If the receiver also cannot determine the frequency  $\omega_0$  accurately, then  $\theta$  may be time varying. Assume that if it is time varying, then it varies slowly enough to be considered constant over one sample interval. Show that the sampled output of the received filter is

$$g_0(t) * f(t) \Big|_{t=0} = \cos \theta. \quad (6.156)$$

Hence, if the phase is correct,  $\theta = 0$ , and the sampled output is unity. However, if the phase error is  $\theta = \pi/2$ , then the sampled output is zero! Hence, uncertainty in  $\theta$  can significantly compromise the performance of the matched filter receiver.  $\square$

The MSK receiver of Figure 6-46 is also coherent, although it permits a 180 degree ambiguity in the carrier phase.

One prime advantage of FSK is the ability to use an incoherent receiver, which does not require knowledge of the carrier phase and can tolerate small inaccuracies in the carrier frequency. The zero-crossing detector of Figure 6-41 is an incoherent receiver, but the matched-filter and correlation receivers are coherent. The question arises whether the matched-filter receiver can be modified to operate incoherently. We will show here that it can, and in Chapter 9 we will show that a slightly more general structure is optimal if the carrier phase is a uniformly distributed random variable over  $(0, 2\pi)$ . Incoherent receivers for FSK perform nearly as well as coherent receivers, so the additional cost of a coherent receiver may not be justified.

We can arrive at an incoherent receiver by heuristic arguments. Matched filters for FSK signals are bandpass filters, although rather crude bandpass filters because of the relatively high sidelobes. Their magnitude frequency response is shown in Figure 6-47. This suggests another intuitive argument for the matched-filter receiver; it simply measures the output of a bank of bandpass filters with center frequencies at each of the signaling frequencies, and selects the largest. Note however that the *magnitude* frequency response will not be affected by phase errors like those in Exercise 6-10. It will also be minimally affected by small frequency errors. This suggests that we could simply estimate the envelope of the filter output, as shown in Figure 6-48. An envelope detector, shown in Figure 6-49, simply finds the amplitude of a sinusoidal

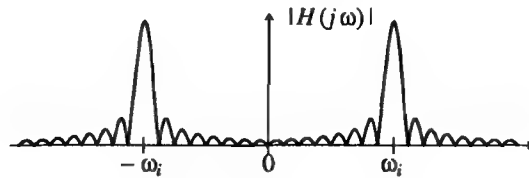


Figure 6-47. The frequency response of a typical matched filter for FSK receivers.

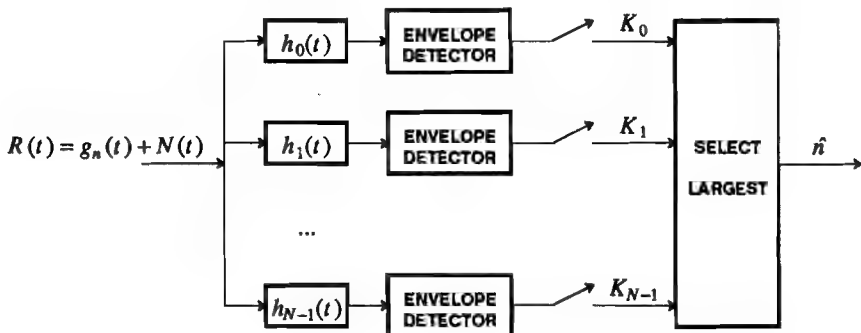


Figure 6-48. An incoherent receiver for isolated pulse FSK signals. The filters  $h_i(t)$  are bandpass filters centered at the signaling frequencies.

signal.

Ideally, all outputs of all the filters in Figure 6-48 but the correct one will be zero. The output of the correct one will be a sinusoid, and the envelope detector will capture the level of the sinusoid. Moreover, unlike the matched filter receiver, the time at which we take the sample at the output of the envelope detector is not critical. The receiver can tolerate relatively large errors in the timing phase, although the average rate at which we take the samples needs to be precise in either case. Timing recovery is covered in Chapter 17.

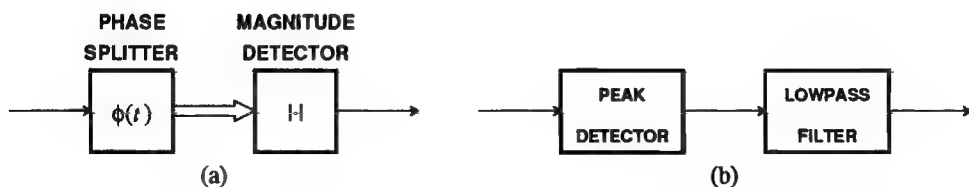
## 6.9. COMBINED PAM AND MULTIPULSE MODULATION

In Section 6.8, we showed that orthogonal multipulse modulation has a maximum spectral efficiency of about one bit/sec-Hz, which is poor in comparison to PAM. The reason that PAM has better spectral efficiency is that increasing the number of bits per symbol does not expand the bandwidth, as it does in orthogonal multipulse. Fortunately, limiting ourselves to pure PAM or orthogonal signaling is not necessary. The two signaling techniques can be combined by choosing a set of  $N$  orthonormal pulse shapes  $\{g_n(t); n = 0, \dots, N-1\}$ , amplitude-modulating each pulse shape with a symbol  $A_{k,n}$  from an alphabet of  $M$  symbols, and linearly combining them, as in

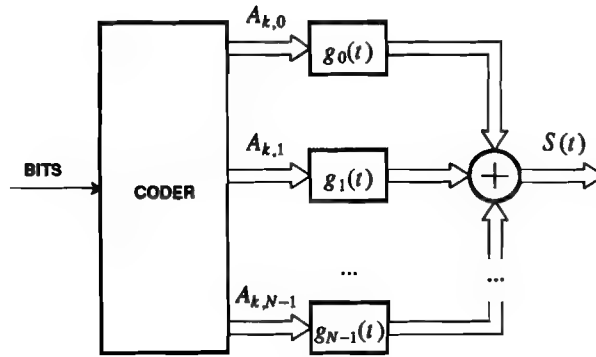
$$S(t) = \sum_{k=-\infty}^{\infty} \sum_{n=0}^{N-1} A_{k,n} g_n(t - kT). \quad (6.157)$$

In each symbol interval of length  $T$ ,  $N$  symbols are simultaneously transmitted using  $N$  distinct pulses, as shown in Figure 6-50. Because the pulse shapes are orthogonal, the superposition of pulses can be sorted out at the receiver by a bank of matched filters. Note that  $S(t)$  can be a complex baseband signal from which we can easily form a passband signal

$$X(t) = \sqrt{2} \operatorname{Re} \{ e^{j\omega_c t} S(t) \}. \quad (6.158)$$



**Figure 6-49.** a. An ideal envelope detector uses a phase splitter to get a complex signal. The magnitude of the phase splitter output is equal to the amplitude of the sinusoidal input (see Problem 6-23). b. An approximate envelope detector uses a peak detector and a lowpass filter.



**Figure 6-50.** A transmitter for combined PAM and orthogonal multipulse.

**Example 6-43.**

PAM is clearly a special case of (6.157) in which  $N = 1$ . For passband PAM (6.157) represents a complex equivalent baseband signal. Interestingly, sometimes the passband PAM signal can be represented directly as a combined PAM and multipulse signal. Consider a passband PAM signal where the carrier frequency  $\omega_c$  has an integer number of cycles in one symbol interval, so  $\cos(\omega_c(t - mT)) = \cos(\omega_c t)$  for any integer  $m$ . Then we can rewrite the quadrature representation (6.48) of a passband PAM signal as

$$X(t) = \sum_{k=-\infty}^{\infty} \text{Re}\{A_k\} g_0(t - kT) + \text{Im}\{A_k\} g_1(t - kT), \quad (6.159)$$

where

$$g_0(t) = \sqrt{2} \cos(\omega_c t) g(t) \quad \text{and} \quad g_1(t) = -\sqrt{2} \sin(\omega_c t) g(t). \quad (6.160)$$

Regardless of the underlying pulse shape  $g(t)$ , the pulses in (6.160) are orthogonal. Hence, for certain carrier frequencies, passband PAM can be viewed as an example of combined PAM and multipulse for  $N = 2$ .  $\square$

**Example 6-44.**

Ordinary orthogonal multipulse signaling is also a special case, where the symbols  $A_{k,n}$  take on values 0 or 1 in such a way that exactly one of the set  $\{A_{k,n}; n = 0, \dots, N-1\}$  has value 1.  $\square$

**Example 6-45.**

MSK (Section 6.8.5) is also a special case. The pulses  $g_0(t)$  and  $g_1(t)$  in Figure 6-45 are amplitude modulated by the symbols  $\pm 1$  or 0. In this case, additional encoding is used to select the symbol sequence  $A_{k,n}$  for each pulse in order to maintain continuous phase.  $\square$

If  $N$  is large, and the symbols  $A_{k,n}$  are appropriately random, then (6.157) will approach a Gaussian distribution, from the central limit theorem. Hence, a key disadvantage of PAM/multipulse is that the peak-to-average power ratio can be larger than with more conventional signals. This suggests that PAM/multipulse modulation is not advisable for either peak-power-limited channels or nonlinear channels. This same

disadvantage, however, can become an advantage when we recall that, as shown in Chapter 4, a signal that achieves channel capacity has a Gaussian distribution.

For  $N$  orthogonal pulses satisfying the generalized Nyquist criterion, the minimum bandwidth is  $B = N/2T$  Hz, where  $T$  is the symbol interval. Thus, the maximum number of orthogonal pulses is  $N = 2BT$ . There are two ways to increase this number, by increasing the symbol interval  $T$ , or by increasing the bandwidth  $B$ . Both approaches are valuable in practice. For example, in multicarrier modulation (explained below) it is usual to hold  $B$  fixed and increase  $T$ ; in code-division multiple access (also explained below) it is usual to hold  $T$  fixed and increase  $B$ .

Before briefly considering the advantages of each approach, we will first show that the spectral efficiency is independent of  $N$ , and equivalent to that of PAM ( $N = 1$ ). Thus, to first order, the advantages of increasing  $N$  (and hence either  $T$  or  $B$ ) can be gained with no impact on spectral efficiency.

### Spectral Efficiency

The symbol set  $\{A_{k,n}; n = 0, \dots, N-1\}$  can assume one of  $M^N$  values, communicating  $\log_2 M^N$  bits of information per symbol interval  $T$ . Thus the best spectral efficiency is

$$\eta = \frac{\log_2 M^N}{BT} = 2 \cdot \log_2 M, \quad (6.161)$$

from (6.8), where we have used the minimum bandwidth  $B = N/2T$  Hz predicted by the generalized Nyquist criterion. The spectral efficiency of PAM/multipulse is approximately equivalent to PAM with the same alphabet size  $M$ , and independent of the dimensionality  $N$ .

For passband signaling, as in (6.158), the bandwidth occupied is twice as great as for the baseband signal. But since the symbols  $A_{k,n}$  in (6.157) are complex, they can carry twice as much information, so the overall spectral efficiency is the same.

### Increasing the Symbol Interval

Suppose we have a fixed channel with bandwidth  $B$ . We are free to increase the dimensionality  $N$  of the signal set if we simultaneously increase the symbol interval  $T$ . Intuitively, we are compensating for the reduced symbol rate by increasing the number of symbols transmitted per symbol interval, thereby keeping the spectral efficiency fixed.

There are two fundamentally different ways to choose the set of orthogonal pulses as we increase  $T$ . One way is to hold the bandwidth of all the pulses fixed at  $B$ , and somehow make them orthogonal (a method for doing this will be discussed in Chapter 8). When we do this, the effects of non-ideal channel transfer functions can be mitigated, because we can make  $T$  so large that any time dispersion on the channel becomes insignificant relative to  $T$ . This effect is easy to visualize intuitively in the time domain, and will be further quantified in the frequency domain in Chapter 8.

The second way to choose orthogonal pulses as  $T$  increases is to make each pulse have bandwidth on the order of  $1/2T$  Hz, satisfying the ordinary Nyquist

criterion, and make them orthogonal by placing them at different center frequencies, spaced  $1/2T$  Hz apart. This was the approach used to design the idealized pulses (6.129) and the Chang pulses (6.132). Again, the effect of this is to mitigate the effects of non-ideal channels, because as  $T$  (and hence  $N$ ) increases, the bandwidth of each pulse decreases (in inverse proportion). This implies that only a portion of the channel transfer function over a narrower and narrower bandwidth affects each pulse transmission. Eventually, for sufficiently large  $T$ , the channel transfer function will be essentially constant over the bandwidth of the pulse, and introduce insignificant ISI. Furthermore, the pulses will tend to retain their orthogonality because they are in approximately non-overlapping frequency bands, both at the channel input and output (the latter assuming the channel is linear and time-invariant). This is the approach taken in multicarrier modulation below, where we will further consider the effects of ISI.

An additional advantage of increasing  $T$  is that impulse noise phenomena that are highly localized in time will have less impact on pulses of greater time duration.

### Increasing Bandwidth

The second way of increasing  $N$  is to hold the symbol interval  $T$  constant and increase the bandwidth  $B$ . This is not feasible on many media, if the additional bandwidth is not available. However, for radio, the medium itself has very broad bandwidth and is typically *frequency-division multiplexed (FDM)* by assigning different users to non-overlapping frequency bands. FDM is closely related to the multicarrier modulation mentioned above, except that in FDM the different users typically do not overlap one another in frequency.

An alternative approach to maintaining the separation of the users is to aggregate  $N$  users sharing a bandwidth  $B$ , where now  $B$  is  $N$  times as large as the nominal bandwidth required by each user in FDM. Then, rather than each user transmitting a pulse that does not overlap the bandwidth of the other users, all users are assigned pulses that occupy the full bandwidth  $B$ . The orthogonality of these pulses is assured by the techniques mentioned above and described further in Chapter 8. This approach to sharing a fixed bandwidth among a set of users, allowing their transmissions to completely overlap in frequency, is called *code-division multiple access (CDMA)*, and is elaborated further below. It is particularly advantageous in *cellular radio* systems, as discussed further in Chapter 18.

### Receiver Design

A correlation receiver for the combined PAM plus multipulse signal, shown in Figure 6-51, is slightly different from the correlation receiver in Figure 6-36. As before, consider only the  $k = 0$  symbol interval (which now carries  $N$  symbols),

$$S(t) = \sum_{n=0}^{N-1} A_{0,n} g_n(t), \quad (6.162)$$

and assume the received signal is corrupted only by white Gaussian noise  $N(t)$ . In particular, the received pulse shapes  $h_n(t)$  are identical to the transmitted pulse shapes. Instead of selecting one of  $N$  candidate pulses as done in Figure 6-36, each

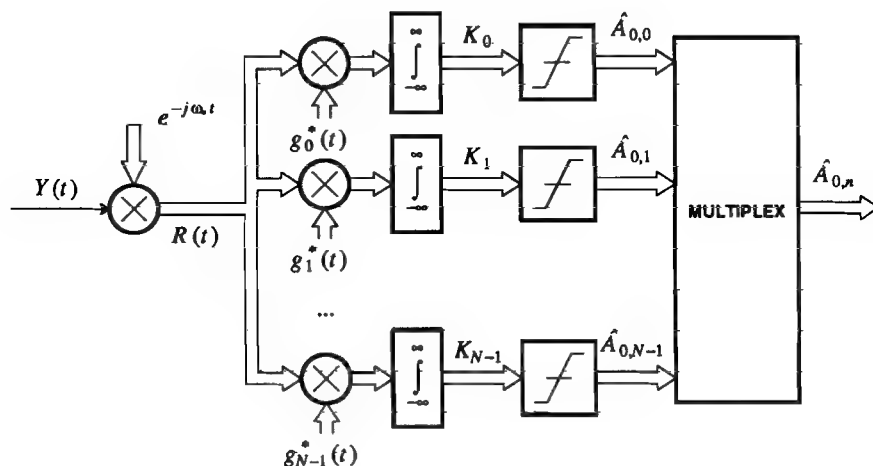


Figure 6-51. A correlation receiver for a combined PAM/multipulse modulation format.

pulse is assumed to carry independent data, and hence has its own slicer, as shown in Figure 6-51. If the pulses satisfy the generalized Nyquist criterion, there is no crosstalk between pulses at the matched filter output, sampled at the appropriate time, so each slicer responds only to its corresponding pulse.

It is not necessary that the  $N$  data symbols be chosen independently, as illustrated by the following example.

#### Example 6-46.

In orthogonal multipulse, only one of the  $N$  pulses is transmitted in each symbol interval. This can be viewed as combined PAM/multipulse where all the symbols but one are set to zero in each symbol interval. We can think of the  $N$  symbols as a vector, where this vector is constrained to have a single unity component, and all the remaining are zero. Thus, the components of the data symbol vector are *not* chosen independently.  $\square$

Where the symbols are not chosen independently, we would not want independent slicers for each pulse, but would rather have one  $N$ -dimensional slicer, as shown in Figure 6-52, that takes account of the dependence of the symbols. The correlation receivers shown in Section 6.8 for orthogonal multipulse are a special case of this design.

### Discrete-Time PAM/Multipulse

The combined PAM and multipulse transmitter of Figure 6-50 and receiver of Figure 6-51 can be quite expensive to implement, particularly if they are implemented in continuous-time, as shown in the figures. Practical implementations, therefore, often implement a simpler transmission system in continuous time, as shown in Figure 6-53b, deriving from it a discrete-time equivalent channel. The combined PAM/multipulse signal is generated in discrete time and transmitted over this discrete-time equivalent channel.



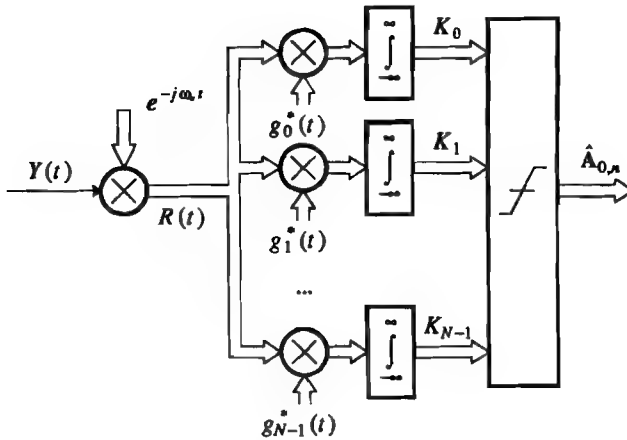


Figure 6-52. A correlation receiver for combined PAM and multipulse where the symbols modulating each pulse are not chosen independently.

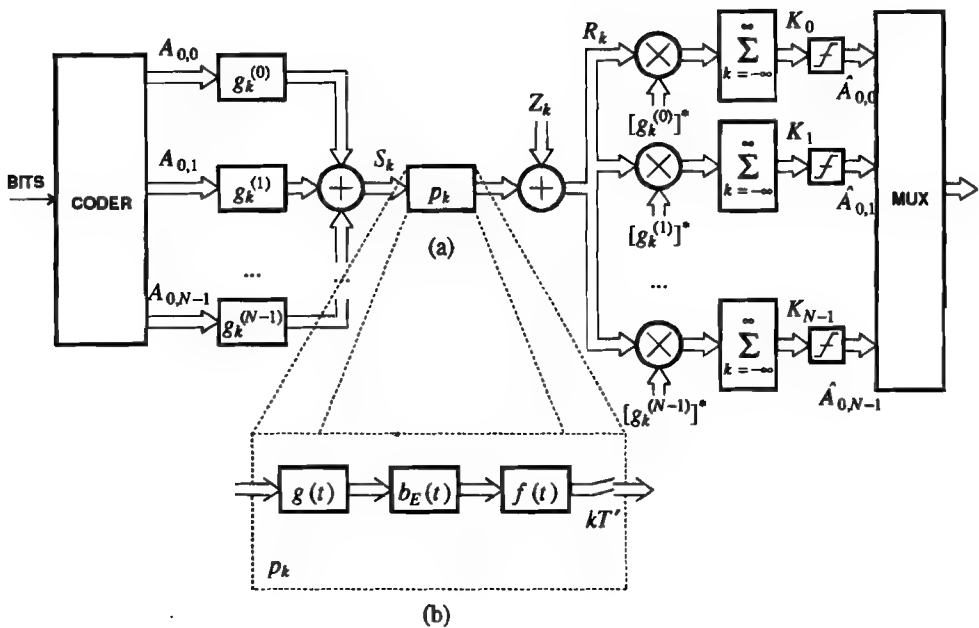


Figure 6-53. (a) A combined PAM/multipulse system where the combination is implemented in discrete-time and transmitted over a discrete-time equivalent channel. The design is shown for an isolated pulse only. (b) The discrete-time equivalent channel can be modeled by a transmit filter, baseband-equivalent channel, baseband-equivalent noise, receive filter, and sampler.

A single symbol interval, or isolated pulse, is given by a discrete-time version of (6.162),

$$S_k = \sum_{n=0}^{N-1} A_{0,n} g_k^{(n)}, \quad (6.163)$$

where  $g_k^{(n)}$  is the  $n$ -th pulse. As before, the pulses  $g_k^{(n)}$  are required to be orthonormal,

$$\sum_{k=-\infty}^{\infty} g_k^{(n)} [g_k^{(m)}]^* = \delta_{n-m}. \quad (6.164)$$

Often, they are chosen to be time-limited, consisting of say a vector of  $K$  samples. In order to have  $N$  orthonormal  $K$ -dimensional vectors, it is necessary that  $K \geq N$ , with a typical choice of  $K = N$ .

A discrete-time correlation receiver for the isolated pulse case, analogous to Figure 6-52, computes the decision variables

$$K_n = \sum_{k=-\infty}^{\infty} R_k [g_k^{(n)}]^*, \quad \text{for } n = 0, \dots, N-1, \quad (6.165)$$

where  $R_k$  is the discrete-time received signal. This is shown in Figure 6-53.

Isolated pulses can be cascaded in time to form a complete signal. In the spirit of (6.157),

$$S_k = \sum_{m=-\infty}^{\infty} \sum_{n=0}^{N-1} A_{mN,n} g_{k-mN}^{(n)}. \quad (6.166)$$

Notice from the subscript of the symbols  $A_{mN,n}$  that the symbol interval is  $N$  times the sample interval. This is intuitive, since we are transmitting  $N$  symbols in one symbol interval. So as not to compromise the robustness of the system, we should transmit  $N$  samples of  $S_k$  per symbol interval. In other words, to transmit  $N$  symbols  $A_{mN,n}$ , for  $n = 0, \dots, N-1$ , we will transmit  $N$  values  $S_k$ , for  $k = mN, \dots, (m+1)N-1$ . If  $T$  is the symbol interval (in seconds) as before, then the sample interval of the discrete-time system will be  $T' = T/N$ .

We will now give two additional examples of combined PAM/multipulse modulation, *multicarrier modulation* and *code-division multiple access* (CDMA). Both of these are commonly implemented in discrete time, but can also be implemented in continuous time.

### 6.9.1. Multicarrier Modulation

Consider forming a PAM/multipulse combination using the pulses

$$g_n(t) = \frac{1}{\sqrt{T}} e^{j\omega_n t} w(t) \quad (6.167)$$

where

$$\omega_n = \frac{2\pi n}{T}; \quad \text{for } n = 0, \dots, N-1, \quad (6.168)$$

and  $w(t)$  is a rectangular windowing function

$$w(t) = \begin{cases} 1; & 0 \leq t < T \\ 0; & \text{otherwise} \end{cases} \quad (6.169)$$

These pulses are similar to the FSK pulses of Example 6-36 with the same frequency separation of  $2\pi/T$  assumed in Example 6-40. This frequency separation ensures continuous phase, but is twice as large as the frequency separation of MSK or the Chang pulses (6.132).

#### Exercise 6-11.

Show that the pulses in (6.167) are orthonormal.  $\square$

Using these pulses,  $S(t)$  in (6.157) consists of superimposed finite-length signals modulated on different carriers. Hence the technique is closely related to frequency division multiplexing, although as with FSK, there is considerable overlap between neighboring frequency bands (see Figure 6-47). The technique is also called *multitone* data transmission. It is somewhat idealized, because the rectangular windowing in (6.167) implies that the signal cannot be bandlimited.

### Discrete-Time Multicarrier

We will consider now a discrete-time version of this. This will allow us to implement the entire bank of transmitter pulse shaping filters and receiver matched filters using the FFT. The resulting transmitter and receiver implementations are cost-effective. Moreover, the discrete-time formulation offers a simple way to combat ISI, as explained below. Define the pulses as

$$g_k^{(n)} = \frac{1}{N} e^{j2\pi nk/N} w_k, \text{ for } n = 0, \dots, N-1, \quad (6.170)$$

where

$$w_k = \begin{cases} 1; & 0 \leq k < N \\ 0; & \text{otherwise} \end{cases} \quad (6.171)$$

An isolated pulse, therefore, is given by (6.163), or

$$S_k = \frac{1}{N} \sum_{n=0}^{N-1} A_{0,n} e^{j2\pi nk/N} w_k. \quad (6.172)$$

Notice that the isolated pulse has duration  $N$ , because the pulses are finite with duration  $N$ . We can apply a discrete-time correlation receiver (6.165) or the matched filter equivalent. This is particularly interesting because the entire bank of correlators or matched filters can be implemented using an efficient FFT implementation.

Observe that  $S_k$  in (6.172) is the  $N^{\text{th}}$  order inverse DFT of  $A_{0,n}$ , for  $n = 0, \dots, N-1$ . This inverse DFT can be efficiently computed using the inverse FFT (IFFT) algorithm, as shown in Figure 6-54. The sequence  $S_k$  is now transmitted over a discrete-time equivalent channel  $p_k$ , where we assume that this channel introduces no ISI, so  $p_k = \delta_k$  (we deal with ISI below). The channel might be

implemented as suggested in Figure 6-53. Assume that since  $p_k = \delta_k$ , the received signal is corrupted only by additive noise  $Z_k$ , so that the received samples are

$$R_k = S_k + Z_k. \quad (6.173)$$

The discrete-time correlation receiver would now compute (6.165), or

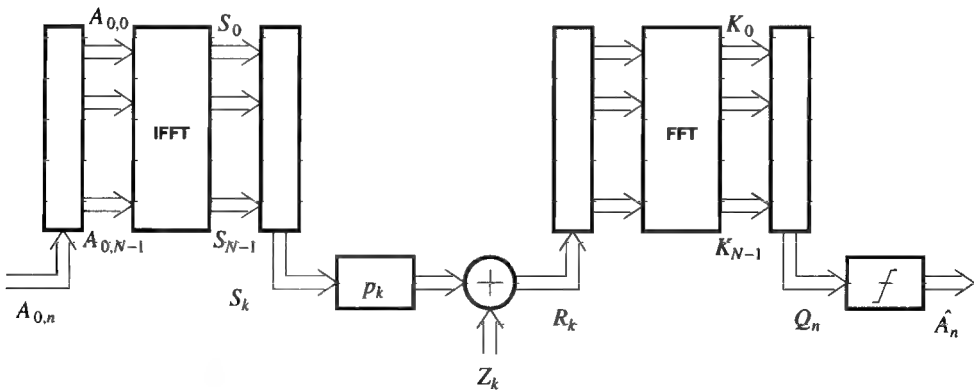
$$K_i = \sum_{k=0}^{N-1} R_k e^{-j2\pi i k/N}, \text{ for } i = 0, \dots, N-1. \quad (6.174)$$

This is the  $N$ -th order DFT, and can be computed using an FFT as shown in Figure 6-54. The entire bank of correlators implied by Figure 6-51 is implemented by a single FFT.

The implementation in Figure 6-54 is an isolated pulse implementation, but it can also handle a stream of pulses. The IFFT and FFT would be computed once per  $N$  samples through the discrete-time equivalent channel, or once per symbol interval. This symbol interval carries a superposition of  $N$  orthogonal pulses and  $N$  symbols.

As with many combined PAM/multipulse systems, since the symbol interval is larger than for a comparable PAM signal, the multicarrier signal is less sensitive to impulse noise (interference that is highly localized in time). Some singular advantages of multicarrier modulation, however, follow from the ability to choose separately the alphabet for each carrier depending on the SNR of the channel at that carrier frequency. In particular, from (6.172) it is clear that for a fixed  $n$ , the sequence of symbols  $A_{k,n}$  modulates the  $n$ -th carrier. By controlling the alphabet for each  $n$ , we can control the transmitted power spectrum and the noise immunity.

- The power spectrum of a multicarrier signal can be dynamically controlled by separately scaling each of the frequency bins. In Chapter 10, we will find a



**Figure 6-54.** A multicarrier modulation system using FFTs. The shaded boxes are parallel-to-serial or serial-to-parallel converters. The discrete-time equivalent channel is expanded into its components at the bottom.

specific shape for the power spectrum that is ideal, in that it permits the performance to approach the theoretical capacity of the channel. Multicarrier signals can easily synthesize that shape.

- It is reasonable to dynamically adjust the choice of alphabet at each carrier frequency as the channel changes over time. Consequently, if a channel has a deep notch in its frequency response, or develops a deep notch due to frequency-selective multipath fading, for example, we would use a very simple binary antipodal alphabet at that frequency. We might even avoid using that frequency altogether. Correspondingly, at frequencies where the channel response is strong, we would use more symbols in the alphabet.

Such power spectrum control is desirable for channels with ISI.

### Multicarrier and ISI

To study the effect of ISI on the multicarrier signal, relax the constraint that  $p_k = \delta_k$  in Figure 6-54, and set the noise  $Z_k$  in Figure 6-54 to zero. Then,

$$R(e^{j\omega T'}) = S(e^{j\omega T'})P(e^{j\omega T'}), \quad (6.175)$$

where  $T' = T/N$  is the sample interval used to transmit the sequential  $S_k$  samples. ( $T = NT'$  is the symbol interval for the entire transmission of (6.172).) Equation (6.175) quantifies the effect of the ISI on the received samples  $R_k$ , but the effect on the decision variables  $K_n$  would be more useful. Since  $K_n$  is the inverse DFT of  $R_k$ , a similar relationship in terms of DFTs rather than DTFTs will precisely quantify this.

Assume that  $p_k$  is zero outside the range  $0 \leq k < M$ , where  $M \leq N$  is some integer. In other words, the number  $N$  of separate carriers is large enough that one symbol interval  $NT' = T$  is longer than the impulse response of the channel. Thus, we can write the noise-free channel output as a finite convolution,

$$R_k = \sum_{i=0}^{N-1} p_i S_{k-i}. \quad (6.176)$$

Let  $w_k$  be the *circular* convolution of  $S_k$  and  $p_k$ ,

$$w_k = \sum_{i=0}^{N-1} p_i S_{(k-i) \bmod N}. \quad (6.177)$$

While the frequency-domain representation of the ordinary convolution (6.176) is (6.175), the frequency domain representation of the circular convolution (6.177) is

$$W_n = A_{0,n} P_n, \quad (6.178)$$

where  $A_{0,n}$  is the DFT (not the DTFT) of  $S_k$  (see Figure 6-54),  $P_n$  is the DFT of  $p_k$ ,

$$P_n = \sum_{k=0}^{N-1} p_k e^{-j2\pi nk/N}, \quad (6.179)$$

and  $W_n$  is the DFT of  $w_k$ . To the extent that a circular convolution is different from the ordinary convolution, (6.178) differs fundamentally from (6.175).

With a modification to the modulation format we can make the circular convolution equal to the ordinary convolution. Suppose that we precede the  $N$  symbols  $S_k$ ,

$0 \leq k < N$ , by  $M$  redundant symbols,

$$S_{-i} = S_{N-i}, \text{ for } 1 \leq i \leq M, \quad (6.180)$$

as shown in Figure 6-55. In this case, (6.177) is equal to the ordinary convolution (6.176). The signals on which we will base our decisions are the DFT of the received signal  $R_k$ , which will now be

$$K_n = A_{0,n} P_n. \quad (6.181)$$

In other words, the  $N$  symbols  $A_{0,n}$  are simply scaled by the  $N$  complex constants  $P_n$  (the DFT of the channel response). For a finite impulse response  $p_k$ , the DFT  $P_n$  is the DTFT sampled at frequencies  $\omega = 2\pi n/N$ . The frequency response of the channel determines the effect of the channel on the decision variables  $K_n$ , as quantified by (6.181).

In most practical situations, we do not know precisely the frequency response of the channel. Often, however,  $P_n$  can be estimated and compensated from observations of  $K_n$ . For instance, if we estimate that  $P_n$  is especially small for some  $n$ , then we might reallocate our encoding so that fewer bits are transmitted on the  $n$ -th carrier. The mechanism used to estimate  $P_n$  is related to the *decision-directed* techniques used for adaptive equalization (Chapter 11) and carrier recovery (Chapter 16). Referring to Figure 6-54, if the decisions  $\hat{A}_n$  are all correct, then we can determine what the variables  $K_n$ , for  $n = 0, \dots, N-1$  should have been by computing an inverse DFT. Comparing what these variables should have been to the actual observation, using (6.181), we obtain an estimate of  $P_n$  for  $n = 0, \dots, N-1$ .

The price paid for this ability to adjust to the channel is the redundant transmission in Figure 6-55 and (6.180), called *cyclic extension* [18]. If  $N$  is large compared to  $M$ , where  $M$  is the length of the impulse response of the discrete-time equivalent channel  $p_k$ , then the overhead associated with transmitting an extra  $M$  symbols per block of  $N$  becomes insignificant. Even without this redundancy, if the impulse

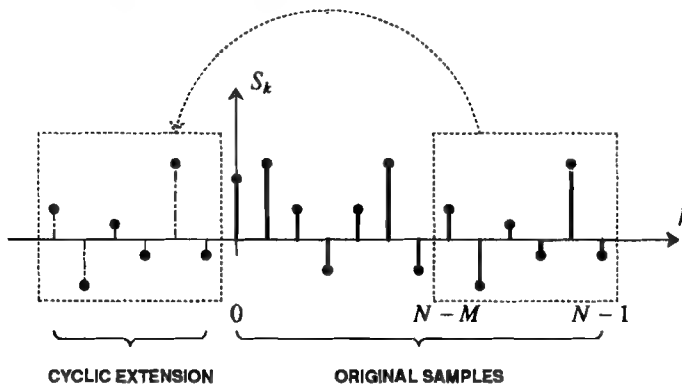


Figure 6-55. For multicarrier transmission, cyclic extension of the samples  $S_k$  simplifies the effect of the channel on the decision variables  $K_n$ .

response of the channel is short compared to the symbol interval  $T$ , the difference between the circular convolution (6.177) and the ordinary convolution (6.176) might be small enough that (6.181) is close enough to be useful [19].

### 6.9.2. Code-Division Multiplexing

Another application of combined PAM and multipulse is *multiple access*, where distinct transmitter-receiver pairs share a single channel. Each transmitter-receiver pair would typically use only one of the orthogonal pulses. As long as the other transmitter-receiver pairs use different orthogonal pulses, the receiver structures given above are effective in separating the signals.

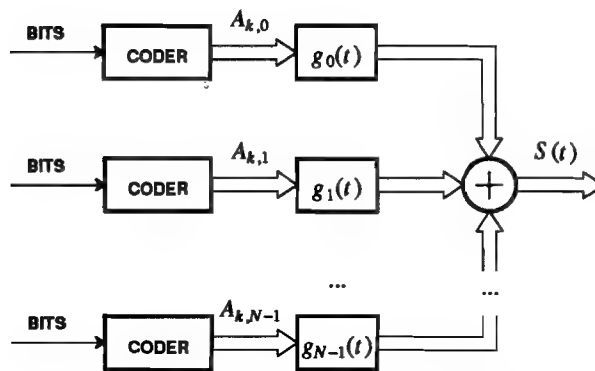
Reversing the order of summation and rewriting (6.157) in the form

$$S(t) = \sum_{n=0}^{N-1} U_n(t), \quad U_n(t) = \sum_{k=-\infty}^{\infty} A_{k,n} g_n(t - kT), \quad (6.182)$$

we can now think of  $S(t)$  as being the superposition of  $N$  PAM subchannel signals  $\{U_n(t), 0 \leq n \leq N-1\}$ , as shown in Figure 6-56. Each of these PAM signals transmits an independent stream of data symbols  $\{A_{k,n}(t), -\infty < k < \infty\}$  using its own distinctive pulse shape  $g_n(t)$ . All these  $N$  PAM signals can share the same channel as long as the pulses they use are orthogonal to one another. The matched filter in the receiver for one subchannel will not respond to the pulse shapes used by the other subchannels, as long as the set of pulses satisfies the general Nyquist criterion.

#### Example 6-47.

In multicarrier modulation, the pulses are chosen to be sinusoids of different frequencies. A multiple access scheme based on this set of pulses would be termed a *frequency-division multiple access (FDMA)* system.  $\square$



**Figure 6-56.** In code-division multiple access (CDMA),  $N$  bit streams share the same channel by using  $N$  distinct orthogonal pulses  $g_i(t)$ .

**Example 6-48.**

An alternative is to choose a set of broadband pulses  $g_n(t)$ , each one of which fills the entire bandwidth of  $|\omega| \leq N \cdot \pi/T$ . For this particular choice of pulses  $g_m(t)$ , (6.182) is known as *code-division multiple access (CDMA)*. This topic is covered more fully in Chapter 18, but suffice it to say that CDMA is but one of several multiple access methods. The pulses used in CDMA are typically generated using a pseudorandom sequence, generated using a technique described in Chapter 12. For now, observe that these pulses can have broad bandwidth (the pseudorandom sequence ensures this) and will be orthogonal to one another.  $\square$

Because the pulses in Example 6-48 individually have much greater bandwidth than would be dictated by the symbol rate, CDMA is related to spread spectrum. In Section 6.7 we described spread spectrum as a technique that can counter certain types of noise and interference signals by greatly expanding the bandwidth of the transmitted pulse. It has very poor spectral efficiency, but can be used in situations where spectral efficiency is not important. In CDMA, each PAM subchannel signal looks like a spread spectrum signal. Unlike in spread spectrum, however, the motivation is not to counter jamming signals so much as to allow other PAM signals (using different orthogonal pulse shapes) to share the same channel. Of course, it is also possible in CDMA to expand the bandwidth beyond  $N \cdot \pi/T$ , thereby gaining both multiple access and immunity to jamming signals of the type discussed in Section 6.7.

## 6.10. OPTICAL FIBER RECEPTION

Optical fiber (Section 5.3) is quite different from other media, and different modulation and receiver techniques are used. The most common optical systems use direct-detection receivers. In direct detection the intensity or power of the light is modulated by a data signal, and the detector converts the received power into an electrical current. Since power is always non-negative, the data signal is non-negative. The simplest case is *on-off keying (OOK)*, where a "zero" is represented by zero intensity and a "one" is represented by positive intensity. Most commercial optical fiber digital transmission systems today use OOK. The technique does not require that the source, LED or laser, be capable of producing a single frequency, which reduces its cost.

The channel from source input  $x(t)$  to detector output  $y(t)$  in Figure 5-18 is a baseband channel, and can be used for baseband PAM transmission, but OOK uses only two levels. Multilevel baseband transmission is normally avoided because at the very high data rates of fiber transmission, the more complex receivers are not justified, given that the symbol rate can be increased easily using better fiber, sources, and detectors. The optical fiber itself has such a high bandwidth that the limitation on bit rate is imposed by the electronics in the transmitter and (particularly) the receiver. Another reason for avoiding multilevel transmission is that it is difficult to control the transmitted power accurately enough.



Since optical fiber with direct detection can be modeled as a baseband PAM channel, we might prematurely conclude that no special treatment is required. However, the noise encountered in optical fiber systems is fundamentally different from that in most other media. In Chapter 8, we analyze the fundamental limits of direct detection, and also introduce a different class of coherent modulation.

## 6.11. MAGNETIC RECORDING

The magnetic recording medium (Section 5.6), like optical fiber, has special properties, and as a result a special form of modulation and detection has evolved. The reading head is differentiating, and hence as seen in Figure 5-40 the output of the channel responds to *transitions* in the write head current rather than pulses as in a conventional channel. As a result, the data is typically encoded as the *presence* or *absence* of a transition in the write waveform, rather than as a positive or negative pulse. This form of modulation is called *NRZI*, and is illustrated in Figure 6-57. Each "one" in the input is translated into a transition, and each "zero" is encoded as no transition. Note that transitions must alternate in sign, so that the direction of a transition has no particular significance.

At the output of the read head, a "one" can be recognized by the presence of a pulse, positive or negative, and a "zero" by the absence of a pulse. The received signal actually has three levels, but both the positive and negative levels have the same meaning, so there is only one bit of information per symbol. The data detection often

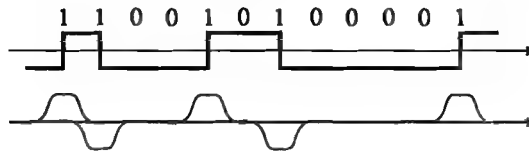


Figure 6-57. The NRZI write waveform and the resulting read voltage.

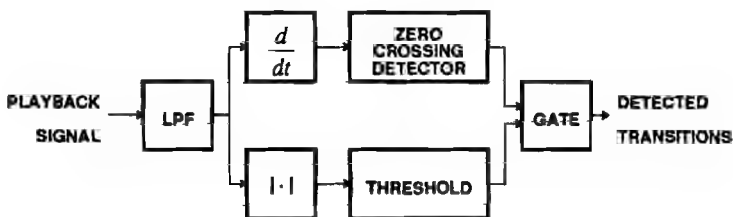


Figure 6-58. A gated peak detector for magnetic recording.

assumes the form shown in Figure 6-58, known as a *gated peak detector* [20]. The playback signal is lowpass filtered to eliminate out-of-band noise and passed through a peak detector (the upper leg). The peak detector consists of a bandlimited differentiator followed by zero-crossing detector. This operation is of course very noisy, and subject to spurious zero-crossings due to noise during the periods of no signal ("zeros", or no transitions, or absence of pulses). To counteract this effect, the lower leg is a pulse validator which ensures that only zero-crossings in the presence of a pulse pass through the gate circuit. The validator consists of a full-wave rectifier and threshold operation to suggest the presence of a pulse. The output of the peak detector is a logical pulse for every valid pulse peak in the read signal. A peak (or zero-crossing) in a particular symbol interval indicates a write transition in the symbol interval, or in other words a "one" bit.

## 6.12. FURTHER READING

More detailed information about FSK can be obtained from Proakis [16] and Lucky, Salz, and Weldon [15], both of which derive the power spectrum of continuous-phase FSK signals. For tutorials on MSK, we recommend Pasupathy [21] and Haykin [22]. The relationship between MSK and QPSK modulation is described in detail by Gronemeyer and McBride [23] and Morais and Feher [24]. For a more general treatment of continuous-phase modulation, see Section 12.4 and the references cited there.

Multicarrier modulation dates back to the 1960s, with early work carried out by Chang [14], Saltzberg [25], Powers and Zimmerman [26], and Darlington [27]. The use of the DFT to construct the transmitted signal is described by Weinstein and Ebert [28] and Hirosaki [19]. Kalet [29] and Ruiz, Cioffi, and Kasturia [30] considered the design of the symbol sets and allocation of bit rates when the channel is distorted, with the latter applying coset coding (Chapter 14). The technique has been applied to the voiceband data channel [31], the high-speed digital subscriber loop (HDSL) channel and the magnetic recording channel [34].

## APPENDIX 6-A MODULATING RANDOM PROCESSES

In this appendix we consider the modulation of a WSS complex-valued random process with a complex-valued carrier. We do this as a sequence of straightforward exercises, thereby highlighting the main results without cluttering the appendix with algebraic manipulation.

Consider a random process defined as

$$Z(t) = S(t)e^{j\omega_c t} \quad (6.183)$$

where  $S(t)$  is a (possibly complex-valued) WSS random process.

**Exercise 6-12.**

Show that if  $S(t)$  is zero-mean, then  $Z(t)$  is WSS and

$$R_Z(\tau) = R_S(\tau)e^{j\omega_c\tau} , \quad (6.184)$$

$$S_Z(j\omega) = S_S(j(\omega - \omega_c)) . \quad (6.185)$$

□

The conclusion of this exercise is that if an equivalent baseband PAM waveform  $S(t)$  is zero-mean and WSS, then the complex-carrier modulated waveform  $Z(t)$  is also WSS. We saw in appendix 3-A that a random phase epoch is required in order for a baseband PAM waveform  $S(t)$  to be WSS.

The WSS random process  $Z(t)$  is complex-valued. For passband PAM it is only the real part that is transmitted, so we need to relate the power spectrum of the real part of a complex-valued random process to the power spectrum of the complex-valued random process. In order to do this, it turns out that we need to first investigate the joint wide-sense stationarity of  $Z(t)$  and its complex-conjugate  $Z^*(t)$ .

**Exercise 6-13.**

Show that  $Z(t)$  and  $Z^*(t)$  are jointly WSS if and only if

$$R_{SS^*}(\tau) = 0 , \quad (6.186)$$

or in words, the baseband signal  $S(t + \tau)$  is uncorrelated with its complex-conjugate  $S^*(t)$  for all delays  $\tau$ . Show further that if (6.186) is satisfied,

$$R_{ZZ^*}(\tau) = 0 . \quad (6.187)$$

□

To gain insight into this condition, we need to investigate the relation  $R_{SS^*}(\tau) = 0$ . Define the real and imaginary parts of  $S(t)$  as

$$S(t) = R(t) + jI(t) . \quad (6.188)$$

**Exercise 6-14.**

Show that  $R_{SS^*}(\tau) = 0$  if and only if

$$R_R(\tau) = R_I(\tau) , \quad (6.189)$$

$$R_{RI}(\tau) = -R_{IR}(\tau) = -R_{RI}(-\tau) . \quad (6.190)$$

In particular, (6.190) requires that  $R_{RI}(0) = 0$ , and hence the real and imaginary parts must be uncorrelated when sampled at the same time. □

The conditions of Exercise 6-14 can be summarized as follows: the power spectrum of the real and imaginary parts must be identical, and the cross power spectral density  $S_{RI}(\tau)$  must be imaginary-valued because  $R_{RI}(\tau)$  is odd.

The next question is whether  $R_{SS^*}(\tau) = 0$  is satisfied for a passband PAM signal. First, (6.189) requires that the real and imaginary parts of the baseband complex-valued PAM signal have identical autocorrelation functions (and hence power spectra). Second, (6.189) requires that the cross-correlation function between real and imaginary parts be an odd function of delay  $\tau$ .

### Exercise 6-15.

For a passband PAM signal, the equivalent baseband PAM waveform is of the form

$$S(t) = \sum_{k=-\infty}^{\infty} A_k h(t - kT + \Theta) \quad (6.191)$$

for a possibly complex-valued pulse shape  $h(t)$  and random phase  $\Theta$ . Assume that  $A_k$  is WSS and independent of  $\Theta$ . Show that a sufficient condition for  $R_{SS^*}(\tau) = 0$  is

$$E[A_k A_m] = 0 \quad (6.192)$$

for all  $k$  and  $m$  and without regard for the distribution of  $\Theta$ . Show further that for (6.192) to be satisfied it is sufficient that the real and imaginary parts of  $A_k$  have the same autocorrelation function and be uncorrelated with one another.  $\square$

Now let us find the power spectrum of the real part of a complex modulated baseband signal. Let

$$X(t) = \sqrt{2}\text{Re}\{Z(t)\} = \frac{1}{\sqrt{2}}[Z(t) + Z^*(t)]. \quad (6.193)$$

### Exercise 6-16.

Show that  $X(t)$  is WSS if and only if  $S(t)$  is zero-mean and WSS and  $R_{SS^*}(\tau) = 0$ . Show further that its autocorrelation function under this condition is

$$R_X(\tau) = \text{Re}\{R_Z(\tau)\} = \text{Re}\{e^{j\omega_c \tau} R_S(\tau)\} \quad (6.194)$$

$\square$

The power spectrum of the real-valued modulated signal is also of interest.

### Exercise 6-17.

Show that if the  $X(t)$  of (6.193) is WSS, its power spectrum is made up of shifted and subtracted versions of the power spectrum of  $S(t)$ ,

$$\begin{aligned} S_X(j\omega) &= 0.5[S_Z(j\omega) + S_Z(-j\omega)] \\ &= 0.5[S_S(j\omega - j\omega_c) + S_S(-j\omega + j\omega_c)] \end{aligned} \quad (6.195)$$

$\square$

To summarize, we have shown that under reasonable conditions on the sequence of data symbols and random phase epoch, the passband PAM signal is WSS. Specifically, those conditions are that the data symbols be WSS and have statistics that satisfy the conditions of Exercise 6-15 and that the baseband PAM waveform

have a uniformly distributed random phase. If these conditions are satisfied, it is not necessary for the carrier to have a random phase. It is important to realize that (6.195) is not valid for any power spectrum  $S_S(j\omega)$ , but only those which satisfy the conditions of Exercise 6-14.

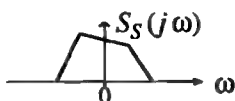
### Exercise 6-18.

Show that if the conditions of Exercise 6-14 are satisfied, then  $S_S(j\omega)$  can be written as

$$S_S(j\omega) = 2S_R(j\omega) - 2j S_{RI}(j\omega), \quad (6.196)$$

where  $S_R(j\omega)$  and  $S_{RI}(j\omega)$  are the power spectra and cross power spectrum of  $R(t)$  and  $I(t)$  in (6.188).  $\square$

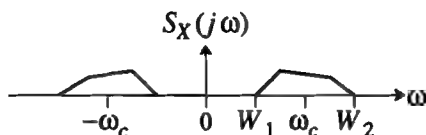
Assume the power spectrum of a complex-valued baseband signal is as shown below:



Using (6.195) we can sketch the power spectrum of

$$X(t) = \sqrt{2} \operatorname{Re} \{ S(t) e^{j\omega_c t} \}. \quad (6.197)$$

Assume  $\omega_c$  is large compared to the bandwidth of  $S(t)$ . The result is shown in the following figure:



The usual  $\sqrt{2}$  factor is used to ensure that the power of  $X(t)$  is the same as the power of  $S(t)$ .

### Exercise 6-19.

Show that when  $X(t)$  is WSS, its power  $R_X(0)$  is the same as the power of  $S(t)$ .  $\square$

## APPENDIX 6-B THE GENERALIZED NYQUIST CRITERION

In this appendix, we first show that a bandwidth of  $N \cdot \pi/T$  is required to satisfy the Nyquist criterion,

$$\frac{1}{T} \sum_{m=-\infty}^{\infty} H_n \left[ j \left( \omega + m \frac{2\pi}{T} \right) \right] H_l^* \left[ j \left( \omega + m \frac{2\pi}{T} \right) \right] = \delta_{l-n}. \quad (6.198)$$

Then we demonstrate a class of practical pulse shapes that satisfy the criterion with a

bandwidth close to the minimum.

### Proof of Necessity

In (6.129) and Figure 6-38 we showed a pulse set with bandwidth  $N\pi/T$  that meets the criterion. This shows that the a bandwidth of  $N\pi/T$  is sufficient. We will now show that it is also necessary.

The left side of (6.198) is a periodic function of  $\omega$  with period  $2\pi/T$ . Consequently, we need only verify (6.198) for  $\omega \in [-\pi/T, \pi/T]$ . (If  $h_n(t)$  is real for all  $n$ , then we need only verify (6.198) for  $\omega \in [0, \pi/T]$ . By conjugate symmetry of  $H_n(j\omega)$  it is automatically satisfied for the rest of the range.) Assume that all pulses  $h_n(t)$  for each  $0 \leq n \leq N-1$  lie within the frequency range  $|\omega| \leq K\pi/T$  for some arbitrary integer  $K$  (we already know that the generalized Nyquist criterion can be met with bandlimited pulses). Then the summation in (6.198) becomes finite, so for  $\omega \in [-\pi/T, \pi/T]$ ,

$$\frac{1}{T} \sum_{m=-M_1}^{M_2} H_n(j(\omega + m\frac{2\pi}{T})) H_l^*(j(\omega + m\frac{2\pi}{T})) = \delta_{l-n}, \quad (6.199)$$

where  $M_1$  and  $M_2$  are integers that depend on  $K$ . Specifically, we want to make them as small as possible while maintaining the equivalence of (6.199) and (6.198). We require that the range  $[\omega - M_1 2\pi/T, \omega + M_2 2\pi/T]$  be at least as large as the total bandwidth of the pulses  $[-K\pi/T, K\pi/T]$  for each  $\omega \in [-\pi/T, \pi/T]$ . If  $K$  is odd, then the smallest values are  $M_1 = M_2 = (K-1)/2$ , so there are  $K$  terms in the summation. If  $K$  is even, we can use different values of  $M_1$  and  $M_2$  in the ranges  $\omega \in [-\pi/T, 0]$  and  $\omega \in [0, \pi/T]$ . For  $\omega \in [-\pi/T, 0]$ , we can use  $M_1 = (K/2) - 1$  and  $M_2 = K/2$ . In the latter range we can use  $M_1 = K/2$  and  $M_2 = (K/2) - 1$ . In all cases, the number of terms in the summation is  $M_1 + M_2 + 1 = K$ .

For each fixed  $\omega$ , define a vector  $\mathbf{H}_n(\omega)$  consisting of the  $K$  terms in the summation,  $H_n(j(\omega + m\frac{2\pi}{T}))$  for  $m = -M_1, \dots, M_2$ . The dimensionality of  $\mathbf{H}_n(\omega)$  is  $K$ . (If real-valued pulses are desired, then  $\omega$  can be restricted to the interval  $\omega \in [0, \pi/T]$  with the constraint  $H_n^*(-j\omega) = H_n(j\omega)$ .) Now we can write (6.199) as

$$\frac{1}{T} \mathbf{H}_n'(\omega) \mathbf{H}_l^*(\omega) = \delta_{l-n}, \quad -\pi/T \leq \omega \leq \pi/T, \quad 0 \leq n, l \leq N-1, \quad (6.200)$$

where  $\mathbf{H}_n'(\omega)$  is the transpose of  $\mathbf{H}_n(\omega)$ . Thus, the generalized Nyquist criterion can be satisfied if, for each  $\omega \in [-\pi/T, \pi/T]$ , a set of  $N$  orthogonal equal-length  $K$ -dimensional Euclidean vectors  $\mathbf{H}_n(\omega)$ ,  $0 \leq n \leq N-1$ , can be found. Clearly,  $N$  orthonormal vectors can be found if their dimensionality is at least  $N$ , or  $K \geq N$ , and cannot be found for smaller  $K$ . Thus, a bandwidth of  $N\pi/T$  will suffice, as confirmed by the earlier example.

We have argued that for  $N$  orthonormal pulses bandlimited to an integer multiple of  $\pi/T$ , the multiple must be  $K \geq N$  to satisfy (6.199). Choosing the minimum value,  $K = N$ , we can now show that the entire bandwidth  $[-N\pi/T, N\pi/T]$  must be used. Thus we prove that the bandwidth cannot be reduced further from  $N\pi/T$ .

Specifically, note that if there is any value of  $\omega$  in this interval where all  $N$  vectors are zero-valued, then (6.200) cannot be true. To see this, define the  $N \times K$  matrix  $\mathbf{H}(\omega)$  where row  $n$  is  $\mathbf{H}_n'(\omega)$ , and note that (6.200) is equivalent to

$$\frac{1}{T} \mathbf{H}(\omega) \mathbf{H}^H(\omega) = \mathbf{I} \quad (6.201)$$

for all  $\omega \in [-\pi/T, \pi/T]$ , where  $\mathbf{H}^H(\omega)$  is the conjugate transpose and  $\mathbf{I}$  is the identity matrix. Hence  $\mathbf{H}(\omega)$  must have full rank. Furthermore, when  $N = K$ , the matrix is square, so this implies that each component of the vectors  $\mathbf{H}_n(\omega)$  must be non-zero for some  $0 \leq n \leq N-1$ , or else  $\mathbf{H}(\omega)$  would have an all-zero column. This in turn implies that  $H_n(j\omega) \neq 0$  for some  $0 \leq n \leq N-1$  for every  $\omega \in [-N\pi/T, N\pi/T]$ . Thus, we arrive at the following theorem:

**Theorem.**

The minimum aggregate bandwidth required to satisfy the generalized Nyquist criterion is  $N\pi/T$ . More precisely, if a set of pulses is constrained to lie within a bandwidth of  $N\pi/T$ , then the pulses collectively fill this bandwidth, leaving no gaps. If the bandwidth is lowpass (it need not be), then for every  $\omega \in [-N\pi/T, N\pi/T]$ ,  $H_n(j\omega) \neq 0$  for some  $0 \leq n \leq N-1$ . Conversely, there exist sets of pulses satisfying the generalized Nyquist criterion with aggregate bandwidth  $N\pi/T$ .  $\square$

For  $N > 1$ , the minimum-bandwidth set of pulses is not unique. To see this, note that if  $\mathbf{H}(\omega)$  satisfies (6.201), then  $\mathbf{U}\mathbf{H}(\omega)$  will also satisfy (6.201) for any *unitary* matrix  $\mathbf{U}$  (see Problem 6-29). (A matrix  $\mathbf{U}$  is unitary if  $\mathbf{U}^{-1} = \mathbf{U}^H$ , where  $\mathbf{U}^H$  is the conjugate transpose.)

For the four pulses shown in Figure 6-38, if we simply number them left to right, then in the range  $\omega \in (0, \pi/T)$ ,

$$\mathbf{H}(\omega) = \sqrt{T} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (6.202)$$

which satisfies (6.201). In the range  $\omega \in (-\pi/T, 0)$  the matrix is similar, but not identical. These pulses, which are non-overlapping in the frequency domain, illustrate the possibilities, but are not practical due to the discontinuity in the frequency response at the band edge. What is needed is a set of pulses that have gradual rolloff, and thus overlap one another in the frequency domain.

### A Practical Pulse Set With Minimum Bandwidth

The pulse set in (6.129) is not practical to implement because the pulses are ideally bandlimited. In Section 6.8.3 we generalized (6.129) to

$$h_n(t) = q(t) \cos((n + 1/2)\pi t/T), \quad (6.203)$$

where  $q(t)$  is a pulse with  $|Q(j\omega)|$  chosen so that  $q(t) * q(-t)$  satisfies the ordinary Nyquist criterion for symbol rate  $1/2T$ , half the desired symbol rate [14]. We allow the bandwidth of  $q(t)$  to be as high as  $\pi/T$ , twice its theoretical minimum,

allowing gradual rolloff in the frequency domain. While the magnitude of  $Q(j\omega)$  is specified so that each pulse satisfies the ordinary Nyquist criterion, we will now show how to select the phase of  $Q(j\omega)$  so that pulses  $h_n(t)$  are mutually orthogonal and satisfy the generalized Nyquist criterion.

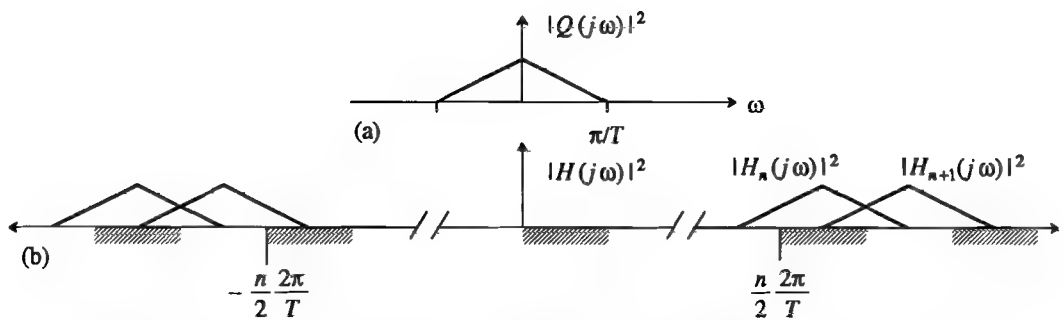
We need to show that (6.198) is satisfied for  $l \neq n$ . Since  $Q(j\omega)$  is bandlimited to  $\pi/T$ ,  $h_n(t)$  and  $h_{n+1}(t)$  have a 50% overlap in the frequency domain, but  $h_n(t)$  and  $h_l(t)$  do not overlap for  $|n - l| \geq 2$ . Thus (6.198) holds trivially for  $|n - l| \geq 2$ .

The only case left is  $l = n+1$ . We will now show that, for any  $q(t) * q(-t)$  satisfying the Nyquist criterion (with respect to symbol rate  $1/2T$ ), the phase of  $Q(j\omega)$  can be chosen such that  $h_n(t)$  and  $h_{n+1}(t)$  satisfy (6.198).  $|H_n(j\omega)|^2$  and  $|H_{n+1}(j\omega)|^2$  are plotted in the Figure 6-59b, where without loss of generality it is assumed that  $n$  is even. The interval  $0 \leq \omega \leq \pi/T$  is highlighted, as are all translates of this interval by  $2\pi/T$  that overlap the spectra in question. We see immediately that there are two cases to consider,  $0 \leq \omega \leq \pi/2T$  and  $\pi/2T \leq \omega \leq \pi/T$ . In the first case, at all translates by  $2\pi/T$ , one or the other of  $H_n(\omega)$  and  $H_{n+1}(\omega)$  is zero, and thus their inner product must be zero. For the second case, both  $H_n(\omega)$  and  $H_{n+1}(\omega)$  have non-zero coordinates for exactly two of the translates by  $kT$ , and for those two translates the vectors are, using the fact that  $Q^*(-j\omega) = Q(j\omega)$  (since  $q(t)$  is assumed real-valued), for the interval  $\pi/2T \leq \omega \leq \pi/T$ ,

$$\mathbf{H}_n(\omega) = [Q^*(j(3\pi/2T - \omega)), Q(j(\omega - \pi/2T))]$$
 (6.204)

$$\mathbf{H}_{n+1}(\omega) = [Q(j(\omega - \pi/2T)), Q^*(j(3\pi/2T - \omega))].$$
 (6.205)

All the zero components of both vectors are omitted, since they will not affect the inner product between the two vectors. The inner product between  $\mathbf{H}_n(\omega)$  and  $\mathbf{H}_{n+1}(\omega)$  will be zero if



**Figure 6-59.** An orthonormal pulse set overlapping in the frequency domain. a. The magnitude-squared of a pulse satisfying the Nyquist criterion at the output of a matched filter, with respect to rate  $1/2T$ . b. The magnitude-squared of two pulses overlapping one another in the frequency domain.



$$\operatorname{Re}\{Q(j(\omega - \pi/2T))Q(j(3\pi/2T - \omega))\} = 0, \quad \pi/2T \leq \omega \leq \pi/T. \quad (6.206)$$

Changing variables, this can be written

$$\operatorname{Re}\{Q(j\omega)Q(j(\pi/T - \omega))\} = 0, \quad 0 \leq \omega \leq \pi/2T. \quad (6.207)$$

This condition can be satisfied by adjusting the phase of  $Q(j\omega)$  to have a particular symmetry about  $\pi/2T$ . An example showing how to do this is included in Section 6.8.3.

## PROBLEMS

- 6-1. Suppose that the rectangular pulse of Example 6-5 is used with the bandlimited channel of Example 6-6, where

$$W = 2\pi/T. \quad (6.208)$$

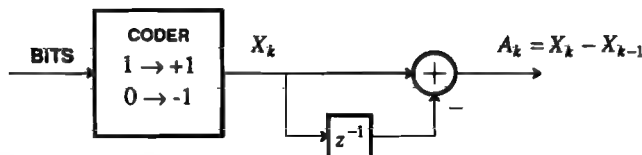
Assume that the symbols  $A_k$  have power (variance)  $E|A_k|^2 = \sigma_A^2$ . If further, assume successive data symbols are uncorrelated. Sketch the power spectrum of the received signal  $R(t)$ . Describe qualitatively the distortion of the signal.

- 6-2. Consider a baseband PAM system using the raised-cosine pulses in (6.24). Assume the symbol sequence is white and normalized, so its power spectrum is

$$S_A(e^{j\omega T}) = 1. \quad (6.209)$$

Show that the transmit power is independent of  $T$  for any roll-off factor  $\alpha$ .

- 6-3. Consider a channel that is ideally bandlimited to  $|\omega/2\pi| \leq 1500\text{Hz}$ . What is the maximum symbol rate using pulses with 50% excess bandwidth? Assume that the receive filter will be an ideal lowpass filter and that ISI is not tolerable.
- 6-4. In an example of a *partial response* system, a symbol sequence is generated as follows:



Assume the incoming bits are random, independent of one another, and zeros and ones are equally probable.

- Find  $S_A(e^{j\omega T})$ , the power spectrum of the symbol sequence. Sketch it.
- Suppose the transmit pulse  $g(t)$  is an ideal lowpass pulse. Find the power spectrum of the baseband PAM transmitted signal. A well-labeled, careful sketch is sufficient.
- Find the pulse  $h(t)$  such that the transmitted signal

$$S(t) = \sum_{m=-\infty}^{\infty} A_m g(t - mT) \quad (6.210)$$

can be written

$$S(t) = \sum_{m=-\infty}^{\infty} X_m h(t - mT). \quad (6.211)$$

Assuming  $g(t)$  from (b), does  $h(t)$  satisfy the Nyquist criterion? Assuming a perfect channel,  $b(t) = \delta(t)$ , is there a stable receive filter  $f(t)$  such that  $p(t) = h(t) * b(t) * f(t)$  satisfies the

Nyquist criterion?

- 6-5. Show that the horizontal eye opening for a pulse with zero excess bandwidth and binary antipodal signaling is zero. Assume there is no coding, so any sequence of symbols is permissible. A consequence of this is that the timing recovery for such a channel would have to be absolutely perfect.
- 6-6. In the baseband PAM receiver of Example 6-18 we can reduce the noise power to arbitrarily low levels by decreasing  $K$ , and hence get arbitrarily good performance. What is wrong with this argument?
- 6-7. In a baseband PAM system, assume  $G(j\omega) = \sqrt{T}$  and  $B(j\omega) = 1$ , so the transmitter sends an impulse stream and the channel has infinite bandwidth. Further assume that the channel noise has power spectrum  $S_N(j\omega) = N_0$ , and assume  $S_A(e^{j\omega T}) = 1$ .
- Show that  $E[|A_k|^2] = 1$ .
  - Show that the power spectrum of the transmitted signal is independent of  $T$ . Hint: Use the results of appendix 3-A, where a random phase is introduced to make the transmit signal WSS.
  - Find the receive filter  $F(j\omega)$  such that the output of the receive filter has a pulse shape with Fourier transform

$$P(j\omega) = T \text{rect}(\omega, \pi/T). \quad (6.212)$$

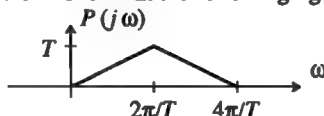
Does this pulse satisfy the Nyquist criterion?

- Find the SNR at the slicer with  $p(t)$  given in part (c).
- Find the SNR at the slicer when the pulse  $p(t)$  has the triangular spectrum of Figure 6-4b, given by

$$P(j\omega) = \begin{cases} T - |\omega|T^2/2\pi; & |\omega| < 2\pi/T \\ 0; & \text{otherwise} \end{cases} \quad (6.213)$$

Compare this SNR to that in part (d).

- 6-8. Suppose the baseband PAM system of Figure 6-1 is designed so that the complex-valued pulse  $p(t)$  has the Fourier transform shown in the following figure:



where  $p(t) = g(t) * b(t) * f(t)$ .

- Does this pulse satisfy the Nyquist criterion? Does  $\text{Re}\{p(t)\}$  satisfy the Nyquist criterion?
- Suppose that the symbols  $A_k$  are uncorrelated, so that

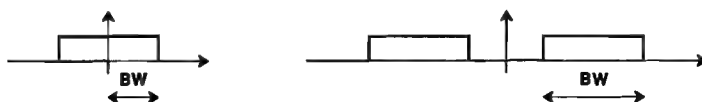
$$S_A(e^{j\omega T}) = \sigma_A^2. \quad (6.214)$$

Assume no noise on the channel,  $N(t) = 0$ . Give the power spectrum of the input to the sampler  $Q(t)$ . Sketch it.

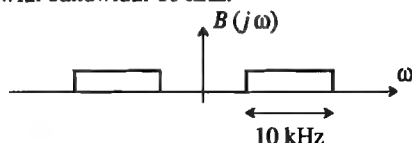
- Find  $p(t)$ .
- 6-9. Consider a channel where the equalized pulse  $p(t) = g(t) * b(t) * f(t)$  is a raised-cosine pulse. Assume  $S_N(j\omega) = N_0$  and  $S_A(e^{j\omega T}) = \sigma_A^2$ . Find the SNR after the receive filter as a function of the excess bandwidth (for the range of zero to 100%) for the following three transmitted pulse shapes and an ideal bandlimited channel:
- The transmitted pulse is an impulse.
  - The transmitted pulse is a raised-cosine pulse with the same excess bandwidth as desired at the receiver.

- (c) The Fourier transform of the transmitted pulse is the square root of the Fourier transform of a raised-cosine pulse (tedious).
- 6-10. Suppose you are to design a digital communication system to transmit a speech signal sampled at 8 kHz with 8 bits per sample. Find the minimum bandwidth required for each of the following methods:
- Binary antipodal baseband PAM.
  - Binary antipodal passband PAM.
  - 4-PSK.
  - 16-QAM.

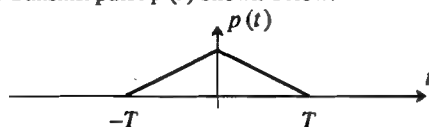
Define bandwidth to cover positive frequencies only, as shown in the following figure:



- 6-11. Consider a channel with bandwidth 10 kHz:



- What is the frequency response  $B_E(j\omega)$  of the baseband equivalent channel? What is the impulse response  $b_E(t)$ ?
- Let  $p(t) = g(t) * b_E(t) * f(t)$ , where  $g(t)$  is the impulse response of the transmit filter and  $f(t)$  is the impulse response of the receive filter. Enter the maximum *bit rate* achievable with zero ISI using the following methods:
  - 4-PSK with  $p(t)$  being a minimum-bandwidth pulse,
  - binary antipodal with  $p(t)$  being a 50% excess-bandwidth raised-cosine pulse,
  - 16-QAM with  $p(t)$  being a 100% excess-bandwidth raised-cosine pulse,
  - 16-QAM with the transmit pulse  $p(t)$  shown below:



- Assuming the 4-PSK signal of part (b), give transfer functions for filters  $g(t)$  and  $f(t)$  (and justify). Assume an additive white Gaussian noise channel.
- 6-12. Derive the Nyquist criterion for a passband PAM channel. Use the results from the baseband case if possible. What is the minimum bandwidth required on the channel?
- 6-13. Consider a *Hilbert transformer*, which is a linear filter with impulse response and transfer function given as

$$h(t) = \frac{1}{\pi t} \quad H(j\omega) = -j \operatorname{sgn}(\omega) = \begin{cases} -j; & \omega > 0 \\ j & \omega < 0 \end{cases} \quad (6.215)$$

Show that if  $x(t) = \cos(\omega_0 t)$  is the input, then  $y(t) = \sin(\omega_0 t)$  is the output. Show further that if  $x(t) = \sin(\omega_0 t)$  is the input, then  $y(t) = -\cos(\omega_0 t)$  is the output. Any sinusoidal input experiences a 90 degree phase change.

6-14. Consider an analytic signal

$$z(t) = \text{Re}\{z(t)\} + j \text{Im}\{z(t)\}. \quad (6.216)$$

- (a) Show that  $\text{Im}\{z(t)\}$  can be obtained from  $\text{Re}\{z(t)\}$  by filtering  $\text{Re}\{z(t)\}$  with the Hilbert transformer of Problem 6-13. In other words,

$$\text{Im}\{z(t)\} = \frac{1}{\pi t} * \text{Re}\{z(t)\}. \quad (6.217)$$

- (b) Show that  $-\text{Re}\{z(t)\}$  can be obtained from  $\text{Im}\{z(t)\}$  using the same filter.

6-15. Show that if  $h(t) = 0$  for  $t > 0$  (i.e.  $h(t)$  is *anticausal*), then the real and imaginary parts are related by a Hilbert transform in the frequency domain

$$\text{Im}\{H(j\omega)\} = \frac{1}{\pi\omega} * \text{Re}\{H(j\omega)\}. \quad (6.218)$$

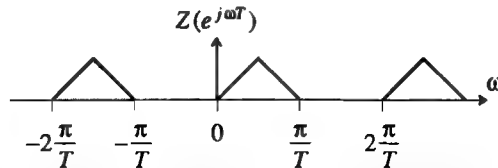
6-16. Consider a discrete-time signal

$$z_k = \text{Re}\{z_k\} + j \text{Im}\{z_k\} \quad (6.219)$$

satisfying

$$Z(e^{j\omega T}) = 0 \quad \text{for } -\pi/T < \omega < 0 \quad (6.220)$$

where  $T$  is the sampling interval. An example is shown in the following figure:



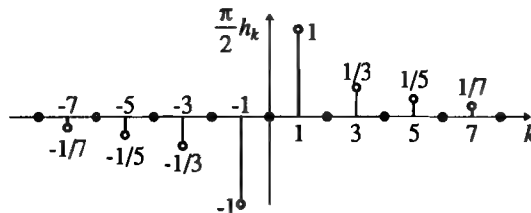
By analogy, such signals are called *discrete-time analytic signals*, although the term "analytic" does not mathematically apply to sequences. Show that  $\text{Im}\{z_k\}$  can be obtained by filtering  $\text{Re}\{z_k\}$  with the *discrete-time Hilbert transformer*

$$H(e^{j\omega T}) = \begin{cases} -j & 0 < \omega < \pi/T \\ j & -\pi/T < \omega < 0 \end{cases}. \quad (6.221)$$

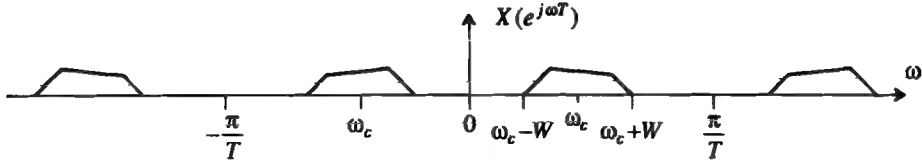
The impulse response is

$$h_k = \begin{cases} 2\sin^2(\pi k/2)/\pi n; & n \neq 0 \\ 0; & n = 0 \end{cases} \quad (6.222)$$

which is shown in the following figure:

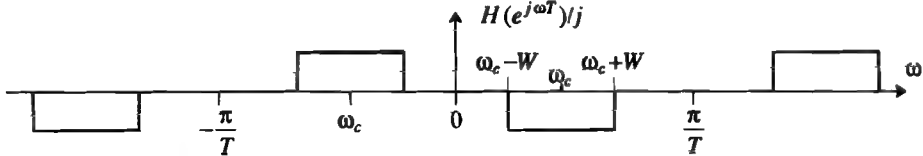


6-17. Assume you are given a discrete-time signal  $x_k$  with sample interval  $T$  and the discrete-time Fourier transform as shown in the following figure:

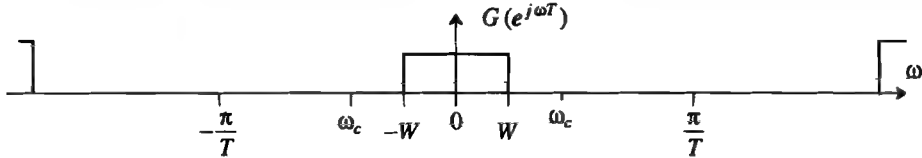


Suppose you are told it is the real part of a discrete-time analytic signal  $z_k$ .

- (a) Show that  $\text{Im}\{z_k\}$  is the result of filtering  $x_k$  with the *real* filter  $h_k$  with frequency response given in the following figure:



- (b) Assume you are given a practical FIR low pass filter  $g_k$  approximating the ideal low pass filter transfer function shown below:



Design a practical filter  $h_k$  approximating the filter in part (a).

6-18. Design a hardware configuration for coders for the constellations in Figure 6-28.

6-19.

- (a) Show that the binary FSK pulses given in (6.138) are orthogonal when

$$\omega_2 - \omega_1 = 2\pi/T \quad (6.223)$$

and  $\omega_1 + \omega_2 = K 2\pi/T$  for some integer  $K$ .

- (b) Show that they are also orthogonal when

$$\omega_2 - \omega_1 = \pi/T \quad (6.224)$$

and  $\omega_1 + \omega_2 = K \pi/T$  for some integer  $K$ .

6-20. Consider the binary CPFSK with pulses given by (6.149) and shown in Figure 6-45c. Assume the nominal carrier frequency satisfies

$$\omega_c = \frac{\omega_0 + \omega_1}{2} = \frac{K\pi}{2T}, \quad (6.225)$$

for some integer  $K$ . Show that the frequency spacing of MSK signals (6.148) is the minimum frequency spacing that results in orthogonal pulses.

6-21. Consider designing an MSK transmission system with  $N = 8$  pulses of different frequencies. Suppose that the lowest frequency is  $f_0 = \omega_0/2\pi = 10$  MHz. Also suppose that the symbol interval is  $T = 1\mu\text{s}$ . Find  $f_1$  through  $f_7$ .

6-22. In this problem we show that an MSK signal can be described as an offset-keyed PSK signal with half sinusoidal pulse shapes.

(a) Show that (6.152) can be written

$$X(t) = \cos(\omega_c t) \sum_{k=-\infty}^{\infty} I_k \sin\left(\frac{\pi t}{2T}\right) w(t - kT) + \sin(\omega_c t) \sum_{k=-\infty}^{\infty} Q_k \cos\left(\frac{\pi t}{2T}\right) w(t - kT), \quad (6.226)$$

where  $Q_k = \cos(\phi_k)$  and  $I_k = b_k \cos(\phi_k)$ . This is close to a passband PAM form, but more work is required.

(b) Show that if  $k$  is even then  $Q_k = Q_{k-1}$ , and if  $k$  is odd then  $I_k = I_{k-1}$ . **Hint:** Use (6.153).

(c) Use parts (a) and (b) to show that

$$X(t) = \cos(\omega_c t) \sum_{k \text{ even}} p(t - kT) (-1)^{k/2} I_k + \sin(\omega_c t) \sum_{k \text{ odd}} p(t - kT) (-1)^{(k+1)/2} Q_k, \quad (6.227)$$

where

$$p(t) = \sin(\pi/2T)(w(t) + w(t - T)) \quad (6.228)$$

is one half of one cycle of a sinusoid. This is a passband PAM signal with pulse shape  $p(t)$  (which extends over  $2T$ ). Notice however that since one of the summations is over even  $k$  and the other is over odd  $k$ , the in-phase and quadrature parts of the signal are offset from one another by  $T$ . The symbol rate is  $1/2T$ , the in-phase symbols are  $(-1)^{k/2} I_k$  for even  $k$ , and the quadrature symbols are  $(-1)^{(k+1)/2} Q_k$  for odd  $k$ .

6-23. Show that if  $A \cos(\omega_c t)$  is fed into an ideal phase splitter to produce a complex output signal, that the magnitude of the complex output signal is the constant  $A/\sqrt{2}$ . Hence, the amplitude of a sinusoid (its *envelope*) can be found using the structure in Figure 6-49a.

6-24. Consider the combined PAM and orthogonal multipulse modulation of (6.157). Suppose you are to achieve a total bit rate of 19.2 kb/s using  $N = 128$  distinct orthonormal pulses. Assume each pulse with modulated using a 4-PSK constellation. Find the symbol interval  $T$ .

6-25.

(a) Show that the DTFT of  $S_k$  in (6.172) is given by

$$S(e^{j\omega T'}) = \frac{1}{N} \sum_{n=0}^{N-1} A_n G_n(e^{j\omega T'}) \quad (6.229)$$

where  $G_n(e^{j\omega T'})$  is the DTFT of the sampled and scaled pulses

$$g_k^{(n)} = e^{j2\pi nk/N} w_k. \quad (6.230)$$

(b) Show that

$$G_n(e^{j\omega T'}) = \frac{\sin\left[\frac{2\pi n - \omega T}{2}\right]}{\sin\left[\frac{2\pi n - \omega T}{2N}\right]} e^{j\frac{N-1}{2N}(2\pi n - \omega T)}. \quad (6.231)$$

**Hint:** the following summation identity may be useful

$$\sum_{k=0}^{N-1} a^k = \frac{1 - a^N}{1 - a}. \quad (6.232)$$

- 6-26. For a possibly complex-valued WSS stationary random process  $S(t)$ , define a cosine-modulated version

$$Z(t) = \sqrt{2} \cos(\omega_c t + \Theta) S(t), \quad (6.233)$$

where  $\Theta$  is uniformly distributed over  $(0, 2\pi)$  and is independent of  $S(t)$ . Show that with this uniformly distributed carrier phase,  $Z(t)$  is WSS and find its power spectrum. Further, show that without the random phase  $Z(t)$  is not WSS, and explain why not.

- 6-27. Show that when a complex carrier  $e^{j(\omega_c t + \Theta)}$  has a random phase  $\Theta$  uniformly distributed on  $[0, 2\pi]$ , the modulated signal (6.193) is WSS for any WSS baseband signal  $S(t)$ .
- 6-28. Let  $Y(t)$  be a zero-mean WSS real-valued random process, and let  $\hat{Y}(t)$  be its Hilbert transform. Show that the signal

$$X(t) = \sqrt{2} \operatorname{Re} \{ e^{j\omega_c t} (Y(t) + j\hat{Y}(t)) \} \quad (6.234)$$

is WSS and find its power spectrum.

- 6-29. We are to design a real-valued set of orthonormal pulses for multipulse modulation. In the range  $0 \leq \omega < \pi/T$ , define the matrix of (6.201) to be

$$\mathbf{H}(\omega) = \frac{\sqrt{T}}{2} \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \quad (6.235)$$

This clearly satisfies (6.201). Sketch  $H_n(j\omega)$  for  $n = 0, 1, 2, 3$ , assuming  $M_1 = 2$  and  $M_2 = 1$  in (6.199). Find expressions for  $h_n(t)$  for  $n = 0, 1, 2, 3$ . How does the bandwidth efficiency compare to that of the pulses in Figure 6-38?

6-30.

- (a) Verify that the  $3 \times 3$  DFT matrix

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{bmatrix} \quad (6.236)$$

satisfies

$$\mathbf{D}\mathbf{D}^H = 3\mathbf{I} \quad (6.237)$$

for all  $\omega$ , where  $\mathbf{I}$  is the identity matrix.

- (b) Use the property in part (a) to find a  $3 \times 3$  matrix  $\mathbf{H}(\omega)$  that satisfies (6.201) for  $\omega \in (-\pi/T, \pi/T)$ .
- (c) Use  $\mathbf{H}(\omega)$  from part (b) to design an orthogonal multipulse pulse set that satisfies the generalized Nyquist criterion. Assume that in defining  $\mathbf{H}(\omega)$ , we use  $M_1 = 0$  and  $M_2 = 2$  in (6.199). Give both time and frequency domain expressions or detailed sketches (whichever is more convenient) for the three pulses.
- (d) Is your pulse set real-valued in time? What is the spectral efficiency if these pulses are used for orthogonal multipulse over a real channel? What is the spectral efficiency if the pulses are used for combined PAM and multipulse, assuming an alphabet size of  $M$ ? How does the spectral efficiency compare to what you would get using the ideal bandlimited pulses of Figure 6-38 for combined PAM and multipulse? How does it compare to ideal baseband and passband PAM? Are the pulses practical?

## REFERENCES

1. T. Noguchi, Y. Daido, and J. Nossek, "Modulation Techniques for Microwave Digital Radio," *IEEE Communications Mag.* 24(10) p. 21 (Oct. 1986).
2. D. Taylor and P. Hartmann, "Telecommunications by Microwave Digital Radio," *IEEE Communications Magazine* 24(8) p. 11 (Aug. 1986).
3. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York (1965).
4. C. R. Cahn, "Combined Digital Phase and Amplitude Modulation Communication Systems," *IRE Trans. on Communications Systems* CS-8 pp. 150-154 (1960).
5. J. C. Hancock and R. W. Lucky, "Performance of Combined Amplitude and Phase-Modulated Communication Systems," *IRE Trans. Communications Systems* CS-8 pp. 232-237 (1960).
6. C. N. Campopiano and B. G. Glazer, "A Coherent Digital Amplitude and Phase Modulation Scheme," *IRE Trans. Communications Systems* CS-10 pp. 90-95 (1962).
7. R. W. Lucky and J. C. Hancock, "On the Optimum Performance of N-ary Systems Having Two Degrees of Freedom," *IRE Trans. Communications Systems* CS-10 pp. 185-192 (1962).
8. R. W. Lucky, *Digital Phase and Amplitude Modulated Communication Systems*, Purdue University, Lafayette, IN (1961).
9. G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "Optimization of Two-Dimensional Signal Constellations in the Presence of Gaussian Noise," *IEEE Trans. on Communications* COM-22(1)(Jan. 1974).
10. G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient Modulation for Band-Limited Channels," *IEEE Journal on Selected Areas in Communications* SAC-2(5)(Sep. 1984).
11. A. Gersho and V. B. Lawrence, "Multidimensional Signal Constellations for Voiceband Data Transmission," *IEEE Journal on Selected Areas in Communications* SAC-2(5)(Sep. 1984).
12. I. S. Gradshteyn and I. M. Ryzhik, "Table of Integrals, Series, and Products," *Academic Press*, (1980).
13. I. Korn, *Digital Communications*, Van Nostrand Reinhold, New York (1985).
14. R. W. Chang, "Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmissions," *Bell System Technical Journal*, (Aug. 1966).
15. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., *Principles of Data Communication*, McGraw-Hill Book Co., New York (1968).
16. J. G. Proakis, *Digital Communications, Second Edition*, McGraw-Hill Book Co., New York (1989).
17. M. L. Doelz and E. H. Heald, "Minimum-Shift Data Communications Systems," *U. S. Patent No. 2,977,417*, (March 28, 1961).
18. A. Peled and A. Ruiz, "Frequency Domain Data Transmission Using Reduced Computational Complexity Algorithms," *IEEE Int. Conf. on ASSP*, pp. 964-967 (April 1980).
19. B. Hirosaki, "An Orthogonal Multiplexed QAM System Using the Discrete Fourier Transform," *IEEE Tr. on Communications*, pp. 982-989 (July 1981).
20. R. Wood, "Magnetic Recording Systems," *IEEE Proceedings* 74(11) p. 1557 (Nov. 1986).
21. S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Communications Magazine* 17(4)(July 1979).
22. S. Haykin, *Communication Systems, 2nd Edition*, John Wiley & Sons, Inc. (1983).



23. S. Gronemeyer and A. McBride, "MSK and Offset QPSK Modulation," *IEEE Trans. on Communications* COM-24(8)(Aug. 1976).
24. D. H. Morais and K. Feher, "Bandwidth Efficiency and Probability of Error Performance of MSK and Offset QPSK Systems," *IEEE Trans. on Communications* COM-27(12)(Dec. 1979).
25. B. R. Saltzberg, "Performance of an Efficient Parallel Data Transmission System," *IEEE Tr. on Communications* COM-15 pp. 805-811 (Dec. 1967).
26. E. Powers and M. Zimmerman, "TADIM — A Digital Implementation of a Multichannel Data Modem," *Proc. of ICC*, (1968).
27. S. Darlington, "On Digital Single-Sideband Modulators," *IEEE Tr. on Circuit Theory* OT-17 pp. 409-414 (Aug. 1970).
28. S. B. Weinstein and P. M. Ebert, "Data Transmission by Frequency Division Multiplexing Using the Discrete Fourier Transform," *IEEE Tr. on Communications* COM-19(5)(Oct. 1971).
29. I. Kalet, "The Multitone Channel," *IEEE Tr. on Communications* COM-37(2) pp. 119-124 (Feb. 1989).
30. A. Ruiz, J. Cioffi, and S. Kasturia, "Discrete Multiple Tone Modulation with Coset Coding for the Spectrally Shaped Channel," *IEEE Tr. on Communications*, (to appear).
31. J. A. C. Bingham, "Multicarrier Modulation for Data Transmission: An Idea Whose Time Has Come," *IEEE Communications Magazine*, pp. 5-14 (May 1990).
32. J. W. Lechleider, "The Optimum Combination of Block Codes and Receivers for Arbitrary Channels," *IEEE Tr. on Communications* COM-38(5)(May 1990).
33. S. Kasturia, J. T. Aslanis, and J. M. Cioffi, "Vector Coding for Partial Response Channels," *IEEE Tr. on Information Theory* 36(4) pp. 741-762 (July 1990).
34. E. Feig and A. Nadas, "Practical Aspects of DFT-Based Frequency Division Multiplexing for Data Transmission," *IEEE Tr. on Communications* 38(7) pp. 929-937 (July 1990).

# 7

---

## SIGNAL and RECEIVER DESIGN

---

In Chapter 6 we described several modulation techniques for both baseband and passband channels. The receiver designs given there were justified on an intuitive basis, with the goal of eliminating ISI at the slicer input and at the same time eliminating as much noise as possible.

In this chapter we develop a much more systematic approach to designing receivers. It will be based on the following idea. Suppose the received signal  $y(t)$  consists of a signal waveform, drawn from a set of known signal waveforms  $s_1(t), s_2(t), \dots, s_L(t)$ , plus additive noise. This will be illustrated by two examples.

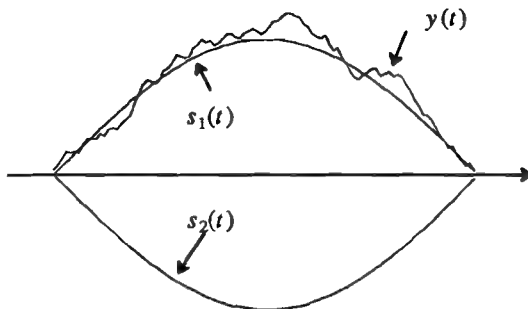
### Example 7-1.

All we know when we design a certain receiver is that the received signal is drawn from a set of two known signals  $s_1(t)$  or  $s_2(t) = -s_1(t)$ , shown in Figure 7-1. That is, we know the shape of the candidate signals, but we do not know which signal is transmitted. The actual received signal  $y(t)$  shown is  $s_1(t)$  plus a small random noise.  $\square$

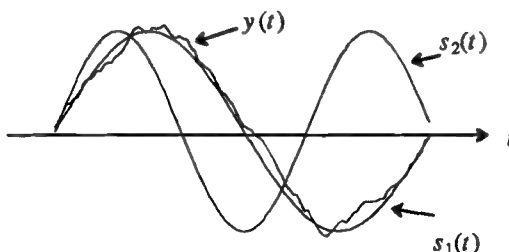
### Example 7-2.

Figure 7-2 shows a similar example, except that the set of two known transmitted signals consists of two different frequencies (binary FSK).  $\square$

In these examples, it is easy to see by looking at the received signal  $y(t)$  which of the signals was actually transmitted. But we need a systematic technique for designing receivers that distinguishes which of the signals was transmitted.



**Figure 7-1.** Two binary antipodal signals,  $s_2(t) = -s_1(t)$ , and a received signal  $y(t)$  that is  $s_1(t)$  plus a small additive noise.



**Figure 7-2.** The set of two known signals is  $s_1(t)$  and  $s_2(t)$  for binary FSK. The received signal  $y(t)$  is actually  $s_1(t)$  plus a small additive noise.

A simple and intuitive approach to receiver design is to form the difference between the received signal  $y(t)$  and each of the signals from the set of known signals.

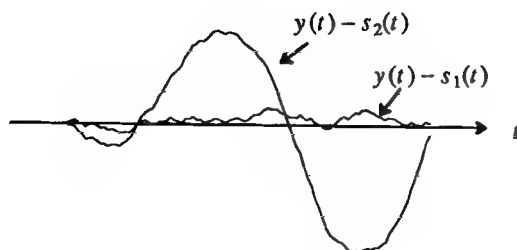
### Example 7-3.

As shown in Figure 7-3 for the example of Figure 7-2, when we form the difference between the received signal  $y(t)$  and  $s_1(t)$ , it is much smaller than if we form the difference between  $y(t)$  and  $s_2(t)$ . Usually this difference signal will be much "smaller" for the actual signal than for the wrong signals. One easy way to quantify the term "smaller" is to calculate the energy of the difference signal, defined as the integral of the square. A receiver following this strategy would calculate

$$\int_{-\infty}^{\infty} |y(t) - s_1(t)|^2 dt, \quad \int_{-\infty}^{\infty} |y(t) - s_2(t)|^2 dt, \quad (7.1)$$

and compare them. The receiver decides that  $s_1(t)$  was transmitted if the first energy is smaller, and  $s_2(t)$  if the second energy is smaller.  $\square$

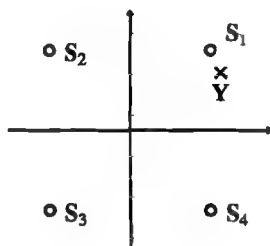
This *minimum-energy* criterion is actually a *minimum-distance* criterion in disguise, in the following sense. In Section 2.6 we showed that signals can be interpreted



**Figure 7-3.** The difference between the received signal and each of the known signals for the signal set and received signal in Figure 7-2.

as vectors in a vector space, and that the notion of inner product and norm (or length) of a vector can be defined. One definition of the length or norm of a continuous-time signal interpreted as a vector is the square-root of the signal energy. The energy of the difference of two signals can thus be interpreted as the square of the distance between the two signal vectors.

When we consider this problem in the context of signal space, the basic idea is a very simple one, and is illustrated in Figure 7-4. The receiver is aware of the set of possible signal waveforms, denoted as vectors by  $\{S_1, S_2, S_3, S_4\}$  in Figure 7-4. The received signal, denoted as a vector  $Y$ , is likely to be close to one of these signal vectors, but not identically equal to that signal since there will be additive noise on the channel, residual errors due to variations in filter transfer functions, etc. On the other hand, we expect that the received signal will be fairly close to one of the known signal waveforms, or else the digital communication system is not likely to work very well. The receiver using a *minimum distance* criterion will choose that signal, from the set of known signals, that is closest to the received signal. In Figure 7-4, a receiver using this minimum-distance criterion will observe which quadrant the received signal falls in, and choose the signal within that quadrant. For the  $Y$  shown, it will choose signal waveform  $S_1$ . (The figure assumes that  $Y$  falls in the subspace spanned by the known



**Figure 7-4.** The idea of minimum-distance receiver design. When the received signal  $Y$  is at the location shown, it is highly suggestive that signal  $S_1$  was received.

signals, but in practice that may not be the case.)

Figure 7-4 looks superficially similar to the signal constellations of Chapter 6, and in fact such constellations are a special case. However, data-symbol constellations are vectors in two-dimensional Euclidean space, while the minimum-distance receiver designs considered in this chapter are applicable to much more general situations, like continuous-time received signals. What is in common among all these applications is the signal-space vector model of the set of known signals.

While we do not explicitly consider the effects of noise in this chapter, this minimum-distance receiver design is primarily motivated by this noise. In Chapter 8, we will determine how to calculate the receiver probability of error for additive Gaussian noise on the channel, when the receiver is designed according to the principles described in this chapter. In Chapter 9, we will show that for additive Gaussian noise, the minimum-distance criterion of this chapter is in fact the "optimal" receiver structure, according to definitions of optimality defined there.

## 7.1. SIGNAL MODEL

In Section 2.6, it was shown how various types of signals can be interpreted as vectors in a vector space, and further how an inner product defines a geometry on this vector space. This geometric interpretation of signals will prove useful in this chapter, because it allows us to solve a number of mathematically equivalent receiver design problems using a common notation and a common solution. Furthermore, it gives us a geometric interpretation of the receiver design, which lends insight.

From the perspective of signal design, the objective is to communicate discrete information by transmitting one signal drawn from a finite and discrete set of possible signals. For most of this chapter, with the exception of Section 7.3, we avoid problems with ISI by considering receiver structures for a detecting a *single* symbol. However, the signal model we describe here is actually much more general than that, as we will see when we consider ISI in Section 7.3.

Accordingly, in a single symbol interval, we assume that the received signal  $\mathbf{Y}$  consists of one of a set of  $L$  possible signals  $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_L\}$  plus some additive noise or error. The receiver does not know *which*  $\mathbf{S}_l$  is received, nor the noise or error. Both  $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_L\}$  and  $\mathbf{Y}$  are assumed to be vectors in an inner product space.

### Example 7-4.

A single sample at the input to the slicer in Section 6.5 is a complex-valued sample  $y$ , which can be considered a vector in two-dimensional real-valued Euclidean space (with two components, the real and imaginary parts of  $y$ ), or as a vector in one-dimensional complex-valued Euclidean space. In this chapter, we will use the complex interpretation. In one-dimensional Euclidean space, the inner product between two such vectors is defined as  $\langle \mathbf{X}, \mathbf{Y} \rangle = xy^*$ .  $\square$

**Example 7-5.**

If the received signal is a complex-valued signal  $y(t)$ , it can be considered an element of an inner product space if

$$\int_{-\infty}^{\infty} |y(t)|^2 dt < \infty . \quad (7.2)$$

In this space, the inner product is defined as

$$\langle X, Y \rangle = \int_{-\infty}^{\infty} x(t)y^*(t) dt . \quad (7.3)$$

□

Our goal is to observe  $Y$ , and use it to detect which of the  $L$  signals was actually transmitted. In the process we have communicated  $\log_2 L$  bits of information.

**Inner Products Used in this Chapter**

For discrete time, we will use the inner product

$$\langle X, Y \rangle = \sum_{k=-\infty}^{\infty} x_k y_k^* \quad (7.4)$$

where the summation may also be finite in a finite-dimensional Euclidean space. For continuous time, we will use the inner product

$$\langle X, Y \rangle = \int_{-\infty}^{\infty} x(t)y^*(t) dt , \quad (7.5)$$

where the infinite integral may be replaced by a finite integral if appropriate. In both cases we will be considering the space of signals for which  $\|X\|^2 = \langle X, X \rangle < \infty$ . Thus, in this chapter, it will be assumed that  $\{S_1, S_2, \dots, S_L\}$  and  $Y$  are finite-energy discrete-time or continuous-time signals. This finite-energy assumption for  $Y$  is problematic if it includes a stationary noise component. However, we will find in Chapter 9 that the receiver design technique of this chapter still applies, although different arguments will be needed.

**The Signal Subspace**

For many of the signal designs considered later, the number of possible signals  $L$  is quite large, and it is advantageous to reduce the complexity of the receiver by going to a two-stage process. This is possible because the dimensionality of the signal set is often much smaller than  $L$ . Let  $M_s$  be the subspace of signal space spanned by the  $L$  signals. Since the number of signals is finite,  $M_s$  must be finite-dimensional; thus, assume it has dimension  $N$ , where  $N \leq L$ . (Note that  $N$  was also defined in Section 6.8 as the dimension of a signal set, in that case a signal set consisting of  $N$  orthonormal signals.) Also choose an orthonormal basis for  $M_s$  consisting of  $\{\Phi_1, \Phi_2, \dots, \Phi_N\}$ , where  $\langle \Phi_i, \Phi_j \rangle = \delta_{i,j}$ . Then the signals can be written in terms of this basis as

$$S_l = \sum_{n=1}^N S_{n,l} \Phi_n \quad (7.6)$$

where the  $S_{n,l}$  are scalar (real or complex) quantities,

$$S_{n,l} = \langle S_l, \Phi_n \rangle. \quad (7.7)$$

#### Example 7-6.

The signals within one symbol interval for PAM combined with orthogonal multipulse modulation are precisely of the form (7.6), where the  $S_{n,l}$  are data symbols, typically chosen independently for different  $n$ .  $\square$

### 7.1.1. Minimum-Distance Receiver Design Criterion

We assume that the received signal is a vector

$$\mathbf{Y} = \mathbf{S}_l + \mathbf{E}, \quad (7.8)$$

for some  $1 \leq l \leq L$ , where the  $l$ -th signal  $\mathbf{S}_l$  is the signal actually transmitted and  $\mathbf{E}$  is some unknown noise or error. Geometrically, our receiver chooses the signal, from among those in the set of known signals  $\{\mathbf{S}_l, 1 \leq l \leq L\}$ , that minimizes  $\|\mathbf{Y} - \mathbf{S}_l\|^2$ , where

$$\|\mathbf{Y} - \mathbf{S}_l\|^2 = \left[ \|\mathbf{Y}\|^2 + \|\mathbf{S}_l\|^2 - 2 \cdot \text{Re}\{ \langle \mathbf{Y}, \mathbf{S}_l \rangle \} \right]. \quad (7.9)$$

Since the term  $\|\mathbf{Y}\|^2$  is independent of  $l$ , it can be ignored, so equivalently the receiver can use the criterion

$$\max_l \left[ \text{Re}\{ \langle \mathbf{Y}, \mathbf{S}_l \rangle \} - \frac{1}{2} E_l \right], \quad E_l = \|\mathbf{S}_l\|^2. \quad (7.10)$$

Here  $E_l$  is the *energy* of the  $l$ -th signal, which is not a function of the received signal  $\mathbf{Y}$  and hence can be pre-computed. The receiver calculates a set of  $L$  real-valued *decision variables*  $\text{Re}\{ \langle \mathbf{Y}, \mathbf{S}_l \rangle \}$ , which are the real part of the inner product of the received signal with the set of known signals.

#### Example 7-7.

The role of the real-part function  $\text{Re}\{ \}$  can be better appreciated by considering the one-dimensional Euclidean case. The real part of the inner product is then

$$\text{Re}\{ \langle \mathbf{X}, \mathbf{Y} \rangle \} = \text{Re}\{ \mathbf{x} \mathbf{y}^* \} = \text{Re}\{ \mathbf{x} \} \cdot \text{Re}\{ \mathbf{y} \} + \text{Im}\{ \mathbf{x} \} \cdot \text{Im}\{ \mathbf{y} \} \quad (7.11)$$

which is equivalent to the inner product in two-dimensional real-valued Euclidean space, where  $\text{Re}\{ \mathbf{x} \}$  and  $\text{Im}\{ \mathbf{x} \}$  are considered to be the two components. Thus, there is a geometric equivalence between one-dimensional complex and two-dimensional real Euclidean space. Taking the real part of the inner product is the key to forming that equivalence.  $\square$

Often  $L$  is quite large, making the number of decision variables large. In this case, the receiver complexity can be reduced considerably by using an  $N$ -dimensional orthonormal basis for the  $N$ -dimensional signal subspace. Substituting for the signal in terms of such a basis, we obtain

$$\operatorname{Re}\{ \langle \mathbf{Y}, \mathbf{S}_l \rangle \} = \operatorname{Re}\left\{ \sum_{n=1}^N S_{n,l}^* \langle \mathbf{Y}, \Phi_n \rangle \right\}. \quad (7.12)$$

The receiver calculates the smaller set of decision variables consisting of inner products,

$$c_n = \langle \mathbf{Y}, \Phi_n \rangle, \quad 1 \leq n \leq N, \quad (7.13)$$

and then uses the decision criterion

$$\max_l \left[ \operatorname{Re}\left\{ \sum_{n=1}^N c_n S_{n,l}^* \right\} - \frac{1}{2} E_l \right]. \quad (7.14)$$

In other words, if the receiver forms the inner product of the received signal with each of the  $N$  orthonormal basis vectors, then it can easily deduce the inner product with each of the  $L$  signals.

It is easily verified that (7.14) is equivalent to

$$\min_l \sum_{n=1}^N |c_n - S_{n,l}|^2, \quad (7.15)$$

which minimizes the distance in  $N$ -dimensional Euclidean space. (Multiplying (7.15) out and throwing away the term that is independent of  $l$  gives precisely (7.14).) Thus, we have shown that, for an  $N$ -dimensional subspace of signals, the minimum-distance receiver design can be recast as a minimum-distance problem in  $N$ -dimensional Euclidean space. To get to that point, the receiver first has to calculate the  $N$  decision variables  $\{c_k, 1 \leq k \leq N\}$ , which are the components of the received signal  $\mathbf{Y}$  in the direction of each of the basis vectors  $\{\Phi_k, 1 \leq k \leq N\}$ . These decision variables summarize the entire received signal, for purposes of calculating the distance.

### Minimum Distance in the Signal Set

If we design receivers according to the minimum-distance criterion, we expect intuitively that signal sets in which the signals are far apart have an advantage over signal sets in which signals are close together. Since the receiver observes which of the possible signals is closest to the received signal, it stands to reason that it is less likely to make an error due to noise or other errors when the other signals are further away.

One important measure of the noise immunity of a given signal set is the *minimum distance between signals*, defined as

$$d_{\min} = \min_{i \neq j} \|\mathbf{S}_i - \mathbf{S}_j\|. \quad (7.16)$$

This minimum distance is the shortest distance between any pair of signals. We will show in Chapter 8 that for Gaussian noise,  $d_{\min}$  is the single most important parameter in predicting the probability of error with a minimum-distance receiver design.



## Equivalence of Passband and Complex Baseband Signals

This subsection will show that the minimum-distance criterion for a passband signal is equivalent to the minimum-distance criterion for the corresponding complex baseband signal. Thus, in this chapter we focus exclusively on complex baseband signals.

If a set of passband signals (all with the same carrier frequency but different pulse shapes)  $\{p_l(t), 1 \leq l \leq L\}$  has corresponding complex baseband signals  $\{s_l(t), 1 \leq l \leq L\}$ , where

$$p_l(t) = \sqrt{2} \cdot \text{Re}\{s_l(t)e^{j\omega_c t}\}, \quad 1 \leq l \leq L, \quad (7.17)$$

then the  $\sqrt{2}$  factor insures that the energy of  $p_l(t)$  is the same as the energy of  $s_l(t)$ . If we apply the minimum-distance criterion directly to a real-valued passband received signal  $y(t)$ , then the receiver calculates the decision variables

$$\begin{aligned} c_l &= \int_{-\infty}^{\infty} y(t)p_l(t) dt = \int_{-\infty}^{\infty} y(t)\sqrt{2} \cdot \text{Re}\{s_l^*(t)e^{-j\omega_c t}\} dt \\ &= \text{Re}\left\{ \int_{-\infty}^{\infty} \left[ y(t) \cdot \sqrt{2}e^{-j\omega_c t} \right] s_l^*(t) dt \right\}. \end{aligned} \quad (7.18)$$

In (7.18) we have used the observation that

$$\text{Re}\{s_l^*(t)e^{-j\omega_c t}\} = \text{Re}\{s_l(t)e^{j\omega_c t}\}. \quad (7.19)$$

Equation (7.18) shows that forming  $y(t) \cdot \sqrt{2}e^{-j\omega_c t}$ , and correlating with the equivalent baseband signal  $s_l(t)$  is equivalent to correlating  $y(t)$  with the passband signal  $p_l(t)$ . Thus the front end of a minimum-distance receiver can consist of a demodulator, with subsequent baseband processing, or one of several equivalent structures considered in Chapter 6.

## 7.2. SPECIFIC MODULATION TECHNIQUES

The model presented in Section 7.1 can be applied in a number of situations. We will give examples in this subsection. The minimum-distance criterion will allow us to easily and systematically *synthesize* receiver structures, rather than having to *assume* receiver structures based on intuition, as we did in Chapter 6. As shown in Chapter 6, the modulation and detection of a passband signal can be considered to be equivalent to detection of a complex baseband signal. Thus, in this chapter, we will assume that all signals are complex valued. The results will apply to complex baseband representations of passband signals, to baseband signals (the special case of real-valued signals) and directly to passband signals (again the real-valued special case).

### 7.2.1. Slicer Design for PAM

In the case of a simple slicer with no ISI, a single input sample to the slicer is of the form

$$y = a_l + e, \quad (7.20)$$

where  $\{a_m, 1 \leq m \leq M\}$  is the complex-valued signal constellation,  $a_l$  is the actual transmitted data symbol, and  $e$  is an unknown error or noise. In this case  $L = M$ , and  $y$  is a complex sample, typically obtained by demodulation, filtering, and sampling.

The received signal  $y$  can be considered to be a vector in a one-dimensional complex Euclidean space, so we associate

$$Y \leftrightarrow y, \quad S_l \leftrightarrow a_l. \quad (7.21)$$

Applying the criterion of (7.9), the slicer chooses the transmitted signal  $\hat{a}_l$  according to

$$\min_l \|Y - S_l\|^2 = \min_l |y - a_l|^2. \quad (7.22)$$

In (7.22), the first  $\|\cdot\|^2$  is a signal-space squared norm, and the second  $|\cdot|^2$  is the squared modulus of a complex number, which is a norm in one-dimensional complex Euclidean space. The slicer thus calculates the distance from the received signal  $y$  in the complex plane, and chooses the signal constellation point that is closest. An equivalent form of the slicer criterion is (7.10),

$$\max_l \left[ 2 \operatorname{Re}\{y a_l^*\} - |a_l|^2 \right] \quad (7.23)$$

#### Example 7-8.

Consider PSK, where the data symbols are of the form  $a_m = e^{j\theta_m}$ . It is also convenient to write the received signal in polar form,  $y = A e^{j\theta}$ . In this case, the  $|a_l|^2$  term is unity for all  $l$ , and hence can be ignored because it is independent of  $l$ . The criterion of (7.23) becomes

$$\max_l \operatorname{Re}\{y e^{-j\theta_l}\} = \max_l A \cos(\theta - \theta_l) \quad (7.24)$$

The receiver thus chooses the angle of the data symbol that is closest to the angle of the received signal, and ignores the magnitude of the received signal. This is obvious from the geometry of the signals in the complex plane. The resulting decision regions are shown in Figure 7-5a for a 4-PSK constellation.  $\square$

#### Example 7-9.

In a QAM signal constellation, the real and imaginary parts of the data symbol are independently modulated, resulting in a rectangular constellation. If  $Q$  is a single slicer input sample (consistent with the notation of Chapter 6), the minimum-distance criterion is

$$\min_l |Q - a_l|^2 = \min_l \left[ (\operatorname{Re}\{Q\} - \operatorname{Re}\{a_l\})^2 + (\operatorname{Im}\{Q\} - \operatorname{Im}\{a_l\})^2 \right]. \quad (7.25)$$

Since both terms are positive, and  $\operatorname{Re}\{a_l\}$  and  $\operatorname{Im}\{a_l\}$  are chosen independently at the transmitter, the sum can be minimized by minimizing the terms individually. Thus, the

complex slicer reduces to two real-valued slicers, one for choosing  $\text{Re}\{a_i\}$  and the other for choosing  $\text{Im}\{a_i\}$ . The decision regions are therefore rectangular. An example is shown in Figure 7-5b for 16-QAM.  $\square$

### Minimum Distance

As pointed out in Section 7.1.1, the minimum distance between all pairs of signals in the known signal set is a characterization of the noise immunity of that signal set. The minimum distance for slicer design is given by

$$d_{\min} = a_{\min}, \quad a_{\min} = \min_{i \neq j} |a_i - a_j|. \quad (7.26)$$

The quantity  $a_{\min}$  is the minimum distance between signal constellation points. This suggests that the constellation should be designed to maximize this minimum distance. This point will be reinforced in Chapter 8, where the actual error probability is considered.

### 7.2.2. Isolated Pulse Reception for PAM: Matched Filter

In Chapter 6 we considered pulse-amplitude modulation, and emphasized the receive filtering that achieved the Nyquist criterion. Let us temporarily ignore inter-symbol interference (ISI) by considering only the reception of a single pulse. Later we will succeed in extending this receiver structure to counter ISI. For an isolated pulse, the received signal will be of the form

$$y(t) = a_l h(t) + e(t) \quad (7.27)$$

where  $a_l$  is the received single data symbol as in Section 7.2.1,  $h(t)$  is the received pulse shape, and  $e(t)$  is some unknown error or noise. In this case, the signal set is one-dimensional, and hence we can choose a single basis signal

$$\phi(t) = h(t)/\sigma_h \quad (7.28)$$

where  $\sigma_h^2$  is the energy of the pulse  $h(t)$ . In the baseband case, all the quantities in (7.27) are real-valued, and in the passband (complex-baseband) case they are complex valued.

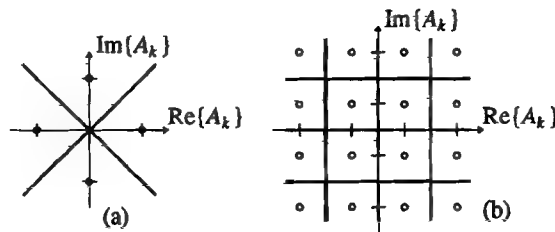


Figure 7-5. The minimum-distance decision regions for the constellations in Figure 6-24.

In order to apply our geometric results,  $y(t)$ ,  $h(t)$ , and  $e(t)$  all have to be square integrable. Associating

$$Y \leftrightarrow y(t), \quad S_l \leftrightarrow a_l \sigma_h \phi(t), \quad (7.29)$$

the receiver criterion of (7.10) becomes

$$\max_l \left[ 2 \operatorname{Re} \left\{ \int_{-\infty}^{\infty} y(t) a_l^* \sigma_h \phi^*(t) dt \right\} - \int_{-\infty}^{\infty} |a_l \sigma_h \phi(t)|^2 dt \right]. \quad (7.30)$$

If we define the decision variable

$$c = \int_{-\infty}^{\infty} y(t) \phi^*(t) dt, \quad (7.31)$$

which is the correlation between the received signal and a normalized version of the known pulse shape, then the criterion becomes

$$\max_l \left[ 2 \operatorname{Re} \{ \sigma_h c a_l^* \} - \sigma_h^2 |a_l|^2 \right], \quad (7.32)$$

which is equivalent to the criterion

$$\min_l |c - \sigma_h a_l|^2. \quad (7.33)$$

Equation (7.33) is equivalent to the minimum distance slicer design for the data symbol  $\sigma_h a_l$  with input sample  $c$ .

Two versions of this receiver structure are shown in Figure 7-6, both of which were displayed earlier in Chapter 6. The correlation receiver is shown in Figure 7-6a, and the matched filter receiver is shown in Figure 7-6b. The matched filter follows from the equivalence of (7.31) to

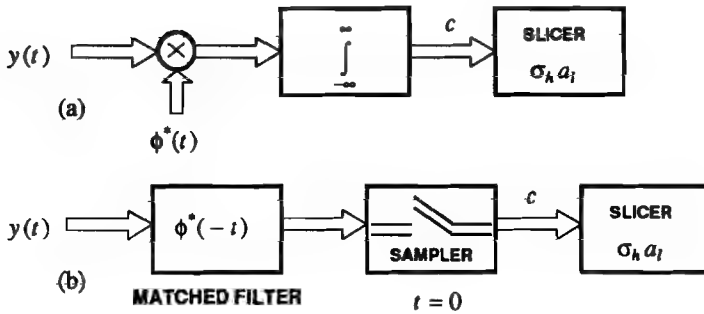
$$c = y(t) * \phi^*(-t) \Big|_{t=0}. \quad (7.34)$$

The matched filter has transfer function  $H^*(j\omega)/\sigma_h$ , and hence does a perfect phase equalization. The Fourier transform of the pulse at the output of the matched filter is  $|H(j\omega)|^2/\sigma_h^2$ , which is non-negative real and has zero phase at all frequencies.

It is useful to define a sampled (at the symbol rate) autocorrelation function for the received pulse  $h(t)$ ,

$$\rho_h(k) = \int_{-\infty}^{\infty} h(t) h^*(t - kT) dt. \quad (7.35)$$

Since  $\rho_h(k)$  is a sampled version of a pulse with Fourier transform  $|H(j\omega)|^2$ , its discrete-time Fourier transform is



**Figure 7-6.** The minimum-distance receiver for an isolated pulse PAM. (a) The correlator realization, and (b) the matched filter realization. The minimum-distance receiver consists of a correlator, or equivalently matched filter and sampler, followed by a slicer.

$$\begin{aligned}
 S_h(e^{j\omega T}) &= \sum_{k=-\infty}^{\infty} \rho_h(k) e^{-j\omega kT} \\
 &= \frac{1}{T} \sum_{m=-\infty}^{\infty} |H(j(\omega + m \cdot \frac{2\pi}{T}))|^2.
 \end{aligned} \tag{7.36}$$

$S_h(e^{j\omega T})$  is called the *folded spectrum* of the received pulse  $h(t)$ . Note that the folded spectrum is real-valued. It will play an important role in Section 7.3, as well as in Chapters 8 and 10, where we consider ISI in detail.

In Chapter 6 we arrived at a receiver structure similar to that shown in Figure 7-6. In that chapter, we had to assume the filter-sampler-slicer structure, and then we optimized it with respect to SNR at the slicer input to arrive at the matched filter (or equivalent correlator). Here this entire receiver was derived from the minimum-distance principle. Beyond giving further justification for the receiver structure assumed in Chapter 6, this principle suggests the appropriate choice for a receive filter. However, the isolated-pulse minimum-distance receiver design ignores the effect of ISI. In general, if we use a matched filter as our receive filter, we will introduce ISI; that is, the pulse shape at the output of the matched filter will *not* obey the Nyquist criterion. This situation will be considered in Section 7.3.

The following two examples illustrate two simple cases where there is ISI, and the folded spectrum is rational (Section 2.5). These examples will be revisited several times in the following chapters.

#### Example 7-10.

Let  $h(t) = \sigma_h \sqrt{2a} e^{-at} u(t)$  where  $u(t)$  is the unit step function,  $a > 0$ , and  $\sigma_h^2$  is the pulse energy. Calculating the pulse autocorrelation function directly, we get

$$\rho_h(k) = \sigma_h^2 \alpha^{|k|}, \quad \alpha = e^{-aT}, \tag{7.37}$$

and the folded spectrum is easily shown to be

$$S_h(z) = \frac{\sigma_h^2(1 - \alpha^2)}{(1 - \alpha z^{-1})(1 - \alpha z)} \quad (7.38)$$

This is a first-order one-pole rational function, with the obligatory second pole at a conjugate-reciprocal location, which forces the folded spectrum to be real-valued.  $\square$

#### Example 7-11.

Let  $h_0(t)$  be a pulse shape that is orthogonal to its translates by  $kT$ . Also assume the energy of  $h_0(t)$  is  $\sigma_0^2$ . Let the actual pulse shape be

$$h(t) = h_0(t) + \alpha h_0(t - T). \quad (7.39)$$

Then the autocorrelation of this pulse is  $\{\dots, 0, \alpha\sigma_0^2, (1 + \alpha^2)\sigma_0^2, \alpha\sigma_0^2, 0, \dots\}$  and the folded spectrum is

$$S_h(z) = \sigma_0^2(\alpha z + (1 + \alpha^2) + \alpha z^{-1}) = \sigma_0^2(1 + \alpha z^{-1})(1 + \alpha z). \quad (7.40)$$

This is a first-order all-zero rational function, again with the second conjugate-reciprocal zero that forces the folded spectrum to be real-valued. The pulse energy is  $\sigma_h^2 = \sigma_0^2(1 + \alpha^2)$ .  $\square$

The Nyquist criterion applied to the output of the matched filter becomes

$$\rho_h(k) = \rho_h(0) \delta_k, \quad S_h(e^{j\omega T}) = \rho_h(0) = \sigma_h^2. \quad (7.41)$$

The pulses in Example 7-10 or Example 7-11 do not satisfy this Nyquist criterion, except for  $\alpha = 0$ . The matched filter does not change the minimum bandwidth required for PAM, since the revised Nyquist criterion of (7.41) still requires a minimum pulse bandwidth of  $\pi/T$  radians/sec ( $1/2T$  Hz).

### Minimum Distance

The distance between signals  $a_i h(t)$  and  $a_j h(t)$  is

$$d^2 = \int_{-\infty}^{\infty} |(a_i - a_j)h(t)|^2 dt, \quad (7.42)$$

or

$$d_{\min} = \sigma_h a_{\min}, \quad (7.43)$$

where  $a_{\min}$  is the minimum distance for the signal constellation. Keeping the signals far apart (improving the noise immunity) is thus equivalent to keeping the minimum distance for the signal constellation large. Not surprisingly, this minimum distance also increases as the pulse energy increases. Only the isolated pulse energy is relevant to the minimum distance, not the shape or other properties of the pulse.

### 7.2.3. Orthogonal Multipulse Modulation

For orthogonal multipulse signaling, the signal set consists of  $N$  orthogonal pulses, each with the same energy  $\sigma_h^2$ , and the received signal is

$$y(t) = \sigma_h \phi_l(t) + e(t), \quad (7.44)$$

where  $\{\phi_n(t), 1 \leq n \leq N\}$  is a set of  $N$  orthonormal waveforms, and  $e(t)$  is some unknown error or noise signal, assumed to be finite-energy.

As shown in Section 7.1.3, the minimum distance receiver forms the set of  $N$  decision variables

$$c_n = \int_{-\infty}^{\infty} y(t) \phi_n^*(t) dt . \quad (7.45)$$

The receiver then calculates the minimum  $N$ -dimensional Euclidean distance between a vector  $\mathbf{c}$  (whose components are the  $c_n$ 's), and the signal vector  $\mathbf{S}_l = [0, 0, \dots, \sigma_h, 0, \dots, 0]$  where the  $\sigma_h$  is in the  $l$ -th position. The minimum-distance criterion recast in Euclidean space is thus

$$\min_l \left[ \sum_{\substack{n=1 \\ n \neq l}}^N |c_n|^2 + |c_l - \sigma_h|^2 \right] = \min_l \left[ \sum_{n=1}^N |c_n|^2 - 2\sigma_h \cdot \text{Re}\{c_l\} + \sigma_h^2 \right] . \quad (7.46)$$

Clearly this is equivalent to the criterion

$$\max_l \text{Re}\{c_l\} . \quad (7.47)$$

The minimum-distance receiver thus correlates the received signal against each of the orthonormal waveforms and chooses the maximum real part of the result. The structure of this receiver is shown in Figure 7-7.

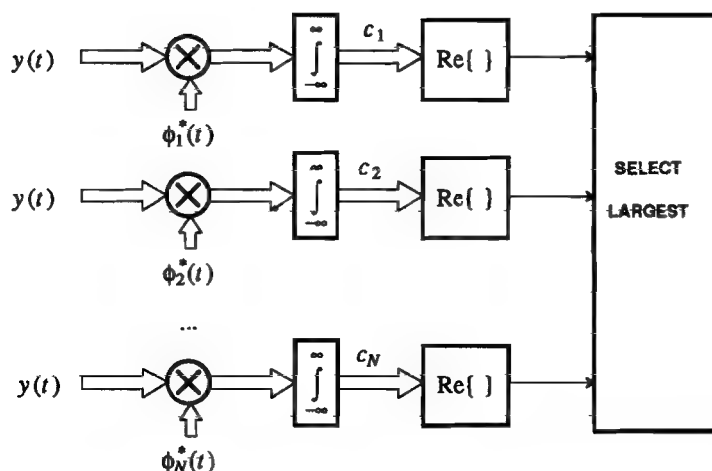


Figure 7-7. An isolated-pulse correlation receiver for orthogonal multipulse transmission.

## Minimum Distance

For orthogonal multipulse, all pairs of signals are equidistant, so the minimum distance is the same as the distance between any pair of distinct signals,  $d_{\min} = \sqrt{2}\sigma_h$ . There are  $N-1$  signals at the minimum distance.

### 7.2.4. Combined PAM and Multipulse

In the case of combined PAM and multipulse, the received signal corresponding to one symbol interval (although this signal is not necessarily time-limited to this interval) is

$$y(t) = \sum_{n=1}^N a_{n,l} \sigma_h \phi_n(t) + e(t), \quad (7.48)$$

where  $\{\phi_n(t), 1 \leq n \leq N\}$  is a set of orthonormal waveforms,  $1 \leq l \leq L$  is an index specifying which signal is transmitted, and  $e(t)$  is a finite-energy unknown error or noise. This is the superposition of a set of orthogonal waveforms, each with the same energy  $\sigma_h^2$  and amplitude modulated by data symbols  $\{a_{n,l}, 1 \leq n \leq N\}$ . This is similar to the general representation for a finite signal set of (7.6). When the  $N$  data symbols are chosen independently from a constellation of size  $M$ , then  $L = M^N$ .

As shown in Section 7.1.1, the receiver first calculates a set of  $N$  decision variables

$$c_n = \int_{-\infty}^{\infty} y(t) \phi_n^*(t) dt, \quad 1 \leq n \leq N, \quad (7.49)$$

and then minimizes the norm in  $N$ -dimensional Euclidean space,

$$\min_l \sum_{n=1}^N |c_n - \sigma_h a_{n,l}|^2. \quad (7.50)$$

The structure of this receiver is illustrated in Figure 7-8.

## Minimum Distance

If we choose two arbitrary vectors of data symbols  $\{a_{n,i}, 1 \leq n \leq N\}$  and  $\{a_{n,j}, 1 \leq n \leq N\}$ , then it is simple to verify that the distance between the corresponding signals, due to the orthonormality of the basis vectors, is

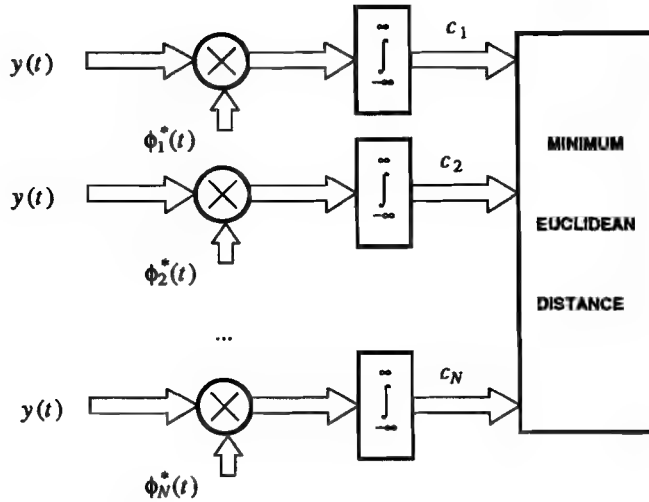
$$d^2 = \sigma_h^2 \sum_{n=1}^N |a_{n,i} - a_{n,j}|^2. \quad (7.51)$$

The minimum distance occurs when  $a_{n,i} \neq a_{n,j}$  for only one value of  $n$ , and thus

$$d_{\min} = \sigma_h a_{\min}, \quad (7.52)$$

the same as for PAM.





**Figure 7-8.** The minimum-distance receiver design for orthogonal multipulse combined with PAM.

### 7.3. PAM WITH INTERSYMBOL INTERFERENCE

Thus far in this chapter we have considered isolated pulses. Conceptually, the minimum-distance receiver design can be used when ISI is present as well. For this case, the received signal is

$$y(t) = \sum_{k=1}^K a_k h(t - kT) + e(t), \quad (7.53)$$

where  $h(t)$  is the received pulse shape, and we make no assumptions about  $h(t)$  being time-limited or satisfying the Nyquist criterion. However, we do assume that  $h(t)$  has finite energy  $\sigma_h^2$  and that  $e(t)$  is a finite-energy unknown error or noise. While we have previously considered the set of signals within one signal interval, in (7.53) we consider the set of all signal sequences of length  $K$ ,  $\{a_k, 1 \leq k \leq K\}$ . Thus, if each data symbol comes from an alphabet of size  $M$ , the entire set of signals in (7.53) has size  $L = M^K$ . By choosing a finite sequence of  $K$  symbols, we ensure that every signal in the set of known signals has finite energy.

This illustrates the generality of the earlier formulation; namely, it can apply to multiple PAM pulses by re-interpreting the concept of "signal" to include the entire sequence of PAM pulses amplitude-modulated by the entire sequence of data symbols  $\{a_k, 1 \leq k \leq K\}$ .

Although the dimensionality of the signal set of (7.53) is  $K$ , expanding this signal in orthonormal functions as was done in the last section is not too useful for ISI. We follow an alternative approach here.

### 7.3.1. Receiver Design

Applying the criterion of (7.10), the receiver chooses the sequence of data symbols that satisfies

$$\max_{\{a_k, 1 \leq k \leq K\}} \left[ 2 \cdot \text{Re} \left\{ \sum_{k=1}^K u_k a_k^* \right\} - \sum_{k=1}^K \sum_{m=1}^K a_k a_m^* \rho_h(m-k) \right] \quad (7.54)$$

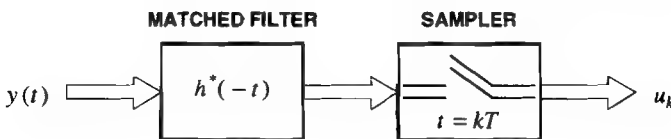
where

$$u_k = \int_{-\infty}^{\infty} y(t) h^*(t - kT) dt, \quad (7.55)$$

and  $\rho_h(k)$  is the pulse autocorrelation function defined in (7.35), where  $\rho_h(0) = \sigma_h^2$ . The samples  $\{u_k\}$  are the sampled output of a filter matched to  $h(t)$ , as in the isolated pulse case, except that now the output is sampled at the symbol rate  $t = kT$  rather than just once at  $t = 0$ . This filter and sampler are known collectively as the *sampled matched filter*, and are illustrated in Figure 7-9. The important feature of this minimum-distance receiver is that the continuous-time received signal  $y(t)$  is turned into a discrete-time received signal  $u_k$ , where the sampling rate is equal to the symbol rate. That discrete-time representation of the received signal is then further processed to make a decision on the entire sequence of symbols  $\{a_k, 1 \leq k \leq K\}$ .

Two observations are in order. First, maximizing (7.54) requires repeating the distance calculation for all  $M^K$  possible symbol sequences  $\{a_k, 1 \leq k \leq K\}$ . Thus, the receiver detects the symbols all at once, instead of doing a symbol-by-symbol detection, which was a major goal in the intuitive receiver design in Chapter 6. It considers all possible sequences of data symbols in order to consider all possible ISI conditions. Second, the receiver designs in Chapter 6 arbitrarily choose a receive filter to eliminate ISI at the slicer input. The minimum-distance criterion chooses a different receive filter that does *not* satisfy the Nyquist criterion, except in the degenerate case where  $\rho_h(k) = \sigma_h^2 \delta_k$ . It then compensates for the resulting ISI in a completely different fashion.

We saw in Chapter 6 that the Nyquist criterion stipulates that the bandwidth of a PAM modulated signal be at least half the symbol rate. Thus, the sampling theorem would dictate a sampling rate *greater* than the symbol rate. The minimum-distance receiver design introduces aliasing in the symbol-rate sampling operation. We did the



**Figure 7-9.** The sampled-matched-filter receiver front end consists of a matched filter followed by a symbol-rate sampler.

same thing in Chapter 6; that is, we chose symbol-rate sampling in order to feed the slicer one sample per symbol. We will see in Chapter 10 that it is common to choose a sampling rate higher than the symbol rate in practice, to address practical concerns.

The minimum-distance receiver design also has the practical problem of high complexity, as manifested by a set of known signals that grows in size exponentially in  $K$ . This is not practical to implement in this form; fortunately, we will find a lower-complexity algorithm, called the Viterbi algorithm, in Chapter 9. Furthermore, in Chapter 10, simpler alternative receiver structures based on equalization will be considered.

### 7.3.2. Equivalent Discrete-Time Criterion

A basic result in Section 2.5, (2.55), provides a factorization of a rational transfer function that is non-negative real on the unit circle. This spectral factorization will now prove useful in deriving a basic minimum-distance receiver structure for ISI. We first consider the special case of no ISI, where this spectral factorization is not needed, and then extend to the general case.

#### Special Case: Orthogonal Pulses

When the successive pulses are orthogonal, or equivalently the pulse shape at the output of the matched filter satisfies the Nyquist criterion ( $\rho_h(k) = \sigma_h^2 \delta_k$ ), (7.54) reduces to

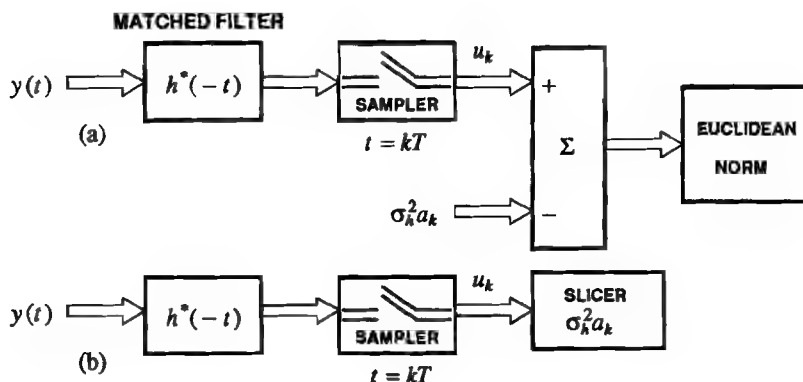
$$\max_{\{a_k, 1 \leq k \leq K\}} \left[ 2 \operatorname{Re} \left\{ \sum_{k=1}^K u_k a_k^* \right\} - \sigma_h^2 \sum_{k=1}^K |a_k|^2 \right]. \quad (7.56)$$

This criterion is equivalent to

$$\min_{\{a_k, 1 \leq k \leq K\}} \sum_{k=1}^K |u_k - \sigma_h^2 a_k|^2, \quad (7.57)$$

which makes intuitive sense, since the output of the sampled matched filter is  $u_k = \sigma_h^2 a_k + e_k$  where  $e_k$  is some unknown error or signal. That is, the signal component at the sampled matched filter output is free of ISI. Equation (7.57) is the minimization of the discrete-time distance between the nominal signal output of the matched filter and the actual sampled matched filter output. This receiver structure is illustrated in Figure 7-10a. The difference between the sampled matched filter output and the normalized sequence of data symbols is formed, and the Euclidean norm in discrete-time is calculated. This is repeated for all possible sequences of data symbols, and the one that minimizes the Euclidean norm is chosen.

The PAM transmitters described in Chapter 6 choose the data symbols independently of one another; that is, each data symbol is chosen from the same alphabet without regard for the other data symbols. In that case, (7.57) can be minimized symbol-by-symbol: since all the terms in the sum are non-negative, the sum is minimized by minimizing each term, where each term is precisely the criterion for a minimum-distance slicer design. Thus, in this case, the simplified structure of Figure 7-10b can be used. Remarkably, this receiver design is precisely the design arrived at in Chapter 6 based on intuitive considerations, except that in this case the receive filter



**Figure 7-10.** The minimum-distance receiver design for PAM when the PAM pulses at the output of the MF satisfy the Nyquist criterion. (a) The receiver that applies to any set of data-symbol sequences, and (b) the special case where data symbols are chosen independently.

is specified as a matched filter. The following example illustrates when the more general structure of Figure 7-10a must be used.

#### Example 7-12.

A simple technique for ensuring no d.c. content in a baseband PAM waveform is a coding technique called *alternate-mark inversion (AMI)*. It is covered in more detail in Chapter 12, but briefly the data symbol  $a_k$  is chosen from an alphabet of size three,  $\{\pm 1, 0\}$  in order to communicate one bit of information. A zero bit is transmitted as  $a_k = 0$ , and a one bit is transmitted by  $|a_k| = 1$ , where the sign of  $a_k$  is chosen to be the opposite sign from the last non-zero  $a_k$ . Since the non-zero  $a_k$  alternate in sign, there is no d.c. content to the sequence of data symbols. If we were to use the simplified structure of Figure 7-10b, the three-level slicer could detect sequences of data symbols that violate the known constraints on the data symbols. For example, it might detect two positive or two negative data symbols in a row. The correct procedure is to use the generalized structure of Figure 7-10a, and calculate the Euclidean norm for valid sequences of data symbols only. In particular, for sequences of  $K$  user bits at the input to the transmitter, there are  $2^K$  valid sequences of data symbols, and not  $3^K$  as might be suggested by the alphabet of size three.  $\square$

This example illustrates a case where the sequence of data symbols embodies *redundancy*; that is, there are restrictions on the sequence of data symbols imposed by the coder that make the number of sequences less than  $M^K$  for an alphabet of size  $M$ . There are many other examples of this, for example for the purpose of controlling the transmitted signal power spectrum (Chapter 12) and for combating noise on the channel (Chapter 14). In each of these cases, the redundancy can be accounted for in the minimum-distance receiver of Figure 7-10a by considering only valid sequences of data symbols. This has the desirable side effect of restricting the number of sequences for which the Euclidean norm must be recalculated.

## General Case

The direct calculation of (7.54) in the general case appears to be impractical from two perspectives:

- Calculating the ML decision variables for just one candidate symbol sequence  $\{a_k, 1 \leq k \leq K\}$  requires on the order of  $K^2$  additions and multiplications. In practical implementations, we can only provide a constant processing rate, or a total computational resource that is proportional to  $K$ . Thus, the calculation of (7.54) is impractical for large  $K$ .
- The total number symbol sequences  $\{a_k, 1 \leq k \leq K\}$  for which (7.54) must be calculated is  $M^K$ . Again, this is impractical to implement.

The special case of  $\rho_h(k) = \sigma_h^2 \delta_k$  yields a receiver structure in which the *continuous-time* minimum-distance receiver is equivalent to a *discrete-time* minimum-distance receiver, with the connection through a sampled matched filter front end. This receiver structure solves the first problem, since the computational overhead for one symbol sequence  $\{a_k, 1 \leq k \leq K\}$  is proportional to  $K$  rather than  $K^2$ . It would be of great significance if this connection to a discrete-time minimum-distance, with its reduction in computational overhead, applied more generally. It does, but we will have to work a bit harder to demonstrate this. Neither the orthogonal pulse nor the general case solves the second problem, that of comparing against  $M^K$  symbol sequences, but a solution to this (the Viterbi algorithm) will be described in Chapter 8.

The fundamental enabling result is a spectral factorization of the folded spectrum. Since the folded spectrum  $S_h(e^{j\omega T})$  is non-negative real-valued, it can be factored in the form (see (2.55)),

$$S_h(z) = A_h^2 G_h(z) G_h^*(1/z^*), \quad (7.58)$$

where  $G_h(z)$  is a monic loosely minimum-phase transfer function. Expressed in the time domain, this spectral factorization is

$$\rho_h(k) = A_h^2 \cdot (g_{h,k} * g_{h,-k}^*). \quad (7.59)$$

Letting  $\{u_k\}$  be the output of the sampled matched filter, as shown in Figure 7-9 and Figure 7-11, define another signal  $\{w_k\}$  as the output of the "precursor equalizer" in Figure 7-11. In other words,  $\{w_k\}$  is the output of a filter  $1/[A_h^2 G_h^*(1/z^*)]$  with input  $\{u_k\}$ . (Generally this will be a stable filter, but there are instances where filters with isolated poles on the unit circle are allowable, so we do not rule out this case.) Expressed in the time domain, this relationship is

$$u_m = A_h^2 w_m * g_{h,-m}^* = A_h^2 \sum_{k=m}^{\infty} w_k g_{h,k-m}^*. \quad (7.60)$$

With these definitions, the continuous-time minimum distance criterion of (7.54) is equivalent to

$$\min_{\{a_k, 1 \leq k \leq K\}} \sum_{m=1}^{\infty} |w_m - \sum_{k=1}^K a_k g_{h,m-k}|^2. \quad (7.61)$$

This criterion (which is similar to, and a generalization of, (7.57)) applies to the

general case where ISI is present at the output of the sampled matched filter, as long as the spectral factorization of (7.58) exists. We can think of this receiver structure as a generalized slicer; the isolated data symbol  $a_k$  is replaced by a filtered sequence of data symbols  $a_k * g_{h,k}$ . Instead of comparing each data symbol independently to a single sample, we compare a filtered sequence of data symbols to a sequence of discrete-time samples  $w_k$ , repeating this comparison for all allowed data-symbol sequences.

For one symbol sequence  $\{a_k, 1 \leq k \leq K\}$ , and where the  $G_h(z)$  is FIR, (7.61) requires a computational load proportional to  $K$ . Thus, as in the orthogonal pulse case, this form of the ML criterion reduces the computation for each candidate symbol sequence from the order of  $K^2$  to  $K$ , but does not address the problem of comparing against  $M^K$  difference symbol sequences. (The latter problem will be successfully addressed in Chapter 9.) If  $G_h(z)$  is not FIR, then it can usually be accurately approximated by an FIR response. In addition to the practical implications of (7.61), it is also of great theoretical interest, because it demonstrates the equivalence of two minimum-distance criteria, (7.54) in continuous time and (7.61) in discrete time. The discrete-time form is much more amenable to further exploration and exploitation in Chapters 8-10.

Showing the equivalence of (7.61) and (7.54) is straightforward. First note that (7.61) can be expressed equivalently as

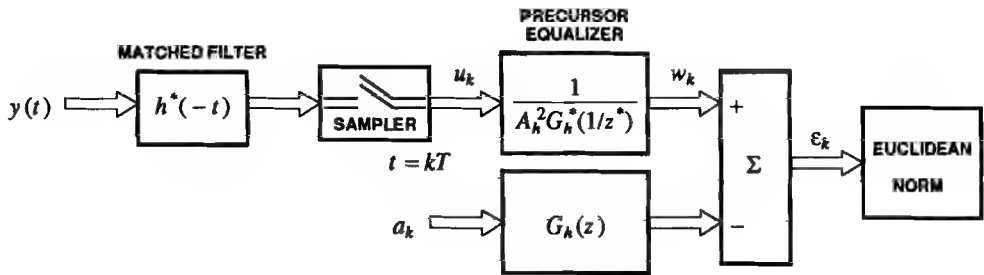
$$\max_{\{a_k, 1 \leq k \leq K\}} \left[ 2 \cdot \sum_{m=1}^{\infty} \operatorname{Re} \left\{ w_m \sum_{k=1}^K a_k^* g_{h,m-k}^* \right\} - \sum_{m=1}^{\infty} \sum_{k=1}^K \sum_{n=1}^K a_k a_n^* g_{h,m-k} g_{h,m-n}^* \right]. \quad (7.62)$$

Exchanging the order of summation, this becomes

$$\max_{\{a_k, 1 \leq k \leq K\}} \left[ 2 \cdot \operatorname{Re} \left\{ \sum_{k=1}^K a_k^* \sum_{m=1}^{\infty} w_m g_{h,m-k}^* \right\} - \sum_{k=1}^K \sum_{n=1}^K a_k a_n^* \sum_{m=1}^{\infty} g_{h,m-k} g_{h,m-n}^* \right]. \quad (7.63)$$

Substituting (7.59) and (7.60) into (7.54), we get precisely the criterion of (7.63) within a multiplicative constant of  $A_h^2$ . The limitation of the summation to  $m \geq 1$  in (7.61) and (7.63) follows from the observation that the earlier terms are not a function of the data symbols, due to the causality of  $g_{h,k}$ .

The receiver structure corresponding to criterion (7.61) is shown in Figure 7-11. As in the special case of Figure 7-10a, it forms the Euclidean norm between two discrete-time signals. One signal,  $\{w_k\}$ , is a filtered version of the sampled matched filter output. The symbol-rate discrete time filter in this upper path is called a *precursor equalizer*. The reason for the terminology "equalizer" is that, as we will see, this filter eliminates a portion (although not all) of the ISI at its input. That is, it inverts (but only partially inverts) the equivalent response of the channel and the sampled



**Figure 7-11.** A minimum-distance receiver for PAM with ISI. This structure is a generalization of Figure 7-10, in that the continuous-time minimum distance criterion is transformed into a discrete-time minimum distance criterion. Unlike Figure 7-10, it applies even in the presence of ISI.

matched filter. It is called a "precursor equalizer" because it eliminates the "anti-causal" or "precursor" response of the channel and sampled matched filter. This terminology will be explained further in Chapter 10.

The second signal used in the discrete-time Euclidean norm is a filtered version of the candidate data-symbol sequence. The filter in this path is an equivalent discrete-time model for the response of the transmit filter, channel, matched filter, and precursor equalizer to the input data symbols, as we will see. The Euclidean distance between precursor equalizer output and the filtered version of the candidate data-symbol sequences is calculated for all possible sequences of  $K$  data symbols. The sequence that minimizes that discrete-time Euclidean distance is chosen. The Euclidean distance must be recalculated many times, once for each allowable sequence of  $K$  data symbols. In the presence of ISI ( $G_h(z) \neq 1$ ) it *never* reduces to a symbol-by-symbol detection as in Figure 7-10b. As in Figure 7-10a, the Euclidean distance should be calculated only for feasible sequences of data symbols, reflecting any redundancy built into the coder at the transmitter.

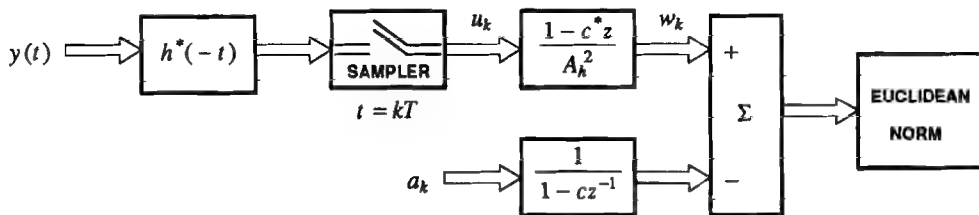
Since  $G_h(z)$  is minimum-phase,  $G_h^*(1/z^*)$  is maximum-phase, and so is  $1/G_h^*(1/z^*)$ . Thus, to be stable this filter must be anticausal, which is impractical, although if it is FIR or can be approximated as FIR, it can be implemented as a causal filter in combination with a delay. (Practical considerations will be addressed in more detail in Chapter 10.)

### Example 7-13.

The lowest-order rational all-pole folded spectrum is

$$S_h(z) = \frac{A_h^2}{(1 - cz^{-1})(1 - c^*z)}, \quad |c| < 1. \quad (7.64)$$

For this case,  $G_h(z) = 1/(1 - cz^{-1})$ , and the resulting receiver structure is shown in Figure 7-12. The precursor equalizer  $(1 - c^*z)/A_h^2$  is an FIR filter. Although it is anti-causal, it is very easily implemented as a causal FIR filter plus a single-sample delay. The candidate data symbols are filtered by a single-pole IIR filter with impulse response  $c^k u_k$  (where in



**Figure 7-12.** The minimum-distance receiver for first-order all-pole folded spectrum.

this case  $u_k$  is the unit-step function).  $\square$

#### Example 7-14.

The lowest-order rational all-zero folded spectrum is

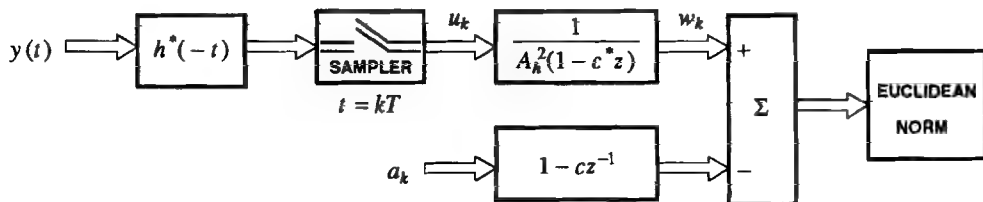
$$S_h(z) = A_h^2(1 - cz^{-1})(1 - c^*z), \quad |c| < 1. \quad (7.65)$$

For this case,  $G_h(z) = (1 - cz^{-1})$ , and the resulting receiver structure is shown in Figure 7-13. The precursor equalizer  $1/(1 - c^*z)$  is an anticausal IIR filter with impulse response  $(c^*)^{-k}u_{-k}$ , and because it is IIR it cannot be implemented directly. Rather, it can only be approximated by an anticausal FIR filter. The candidate data sequence is filtered by a filter with impulse response  $\delta_k - c\delta_{k-1}$ .  $\square$

We will now further motivate the reasons for the equivalence of (7.61) and (7.54). When the input signal is given by (7.53), the output of the sampled matched filter is

$$u_k = \sum_{m=1}^K a_m \rho_h(k - m) + e_k = a_k * \rho_h(k) + e_k, \quad (7.66)$$

for some unknown noise or error  $e_k$ . The signal portion of this output is represented by an equivalent discrete-time filter with impulse response  $\rho_h(k)$  (transfer function  $S_h(z)$ ) and input  $a_k$ . Consider what happens if we put the sampled matched filter output  $u_k$  through a precursor equalizer with transfer function  $1/[A_h^2 G_h^*(1/z^*)]$  as shown in Figure 7-11. The overall transfer function to data symbols is



**Figure 7-13.** The minimum-distance receiver for a first-order all-zero folded spectrum.



$$\frac{S_h(z)}{A_h^2 G_h^*(1/z^*)} = G_h(z). \quad (7.67)$$

and hence the output will be

$$w_k = \sum_{m=1}^K a_m g_{h,k-m} + e_k' \quad (7.68)$$

for some error or noise signal  $e_k'$ . This output signal has the desirable property — heavily exploited in Chapters 9 and 10 — that the resulting overall discrete-time channel is causal. The precursor equalizer has thus removed the anticausal portion of the ISI; that is, the equivalent response up to the precursor equalizer input has a two-sided impulse response, and the precursor equalizer output turns this into a causal impulse response. In Figure 7-11, the actual sequence  $w_k$ , corrupted by noise or error, is compared to what it would be for a candidate sequence of data symbols, (7.68), using a Euclidean distance measure.

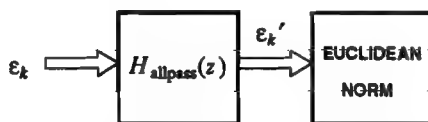
Observe that the norm-squared in (7.61) is calculated for  $1 \leq m \leq \infty$ , even though the sequence of data symbols is finite. This is because the ISI may be infinite in extent, and thus the entire received signal sequence is considered by the minimum-distance criterion.

Another interesting and useful interpretation of the receiver structure of Figure 7-11 is derived in Problem 7-8. In particular, the equivalence between distance in continuous time and discrete time is shown to have a simple geometrical interpretation.

### Non-Uniqueness of the Discrete-Time Criterion

We will now show that the minimum-distance receiver structure that substitutes a discrete-time Euclidean distance for a continuous-time Euclidean distance is not unique. The discrete-time channel model in (7.68) (ignoring for a moment the error signal) is minimum-phase, because of the minimum-phase spectral factorization. However, the spectral factorization of  $S_h(z)$  need not be minimum-phase. There are many front-end discrete-time filters that are equivalent in the sense that they result in the same detected sequence of data symbols  $\{a_k, 1 \leq k \leq K\}$ .

This is illustrated in Figure 7-14, where the error sequence  $\{\varepsilon_k\}$  defined in Figure 7-11 is filtered by an arbitrary rational allpass filter before the Euclidean norm is



**Figure 7-14.** Introducing a rational allpass filter  $H_{\text{allpass}}(z)$  before the Euclidean norm in Figure 7-11.

calculated. By Parseval's theorem, since the allpass filter has a unit-magnitude Fourier transform, it does not change the Euclidean norm, i.e.

$$\sum_{m=1}^{\infty} |\epsilon_m|^2 = \sum_{m=1}^{\infty} |\epsilon'_m|^2, \quad (7.69)$$

as long as we assume that  $\epsilon_m = 0$  for  $m \leq 0$ . Hence, the allpass filter will not change the sequence of data symbols chosen by the receiver. Now, we can move the allpass filter through the summation in Figure 7-11, replacing the filter  $1/A_h^2 G_h^*(1/z^*)$  by  $H_{\text{allpass}}(z)/A_h^2 G^*(1/z^*)$  and replacing  $G_h(z)$  by  $H_{\text{allpass}}(z)G_h(z)$ . This replacement has no effect on the data-symbol sequence chosen by the receiver. If this allpass filter has all poles inside the unit circle (and hence all zeros outside), then it is a stable causal filter, and does not destroy the causality of the channel model in (7.68) either. Effectively, this changes the discrete-time channel model from minimum-phase to non-minimum-phase. While this change would not appear to be harmful, it is shown in Problem 2-22 that a causal minimum-phase sequence has the property that, among all sequences with the same Fourier transform magnitude, it is *maximally concentrated* near zero delay. Thus, in this sense the impulse response of the minimum-phase channel model has minimum intersymbol interference, among all impulse responses with the same Fourier transform magnitude. Stating this another way, Problem 2-23 shows that the impulse response of the filter  $G_h(z)$  is more concentrated near the origin than the impulse response of the modified channel model  $H_{\text{allpass}}(z)G_h(z)$ , and thus has less ISI. While this property does not affect the minimum-distance receiver, in the sense that it chooses the same data-symbol sequence, it is very important for another practically important receiver structure based on a similar decomposition (the decision-feedback precursor equalizer of Chapter 10).

### Minimum Distance

The minimum distance between known signals was asserted in Section 7.2 to be a measure of the noise immunity of the modulation technique. This minimum distance for PAM with ISI is obtained by considering two different sequences of data symbols  $\{\tilde{a}_k, 1 \leq k \leq K\}$  and  $\{\hat{a}_k, 1 \leq k \leq K\}$ . Letting  $\epsilon_k$  be the difference between these two sequences,

$$\epsilon_k = \tilde{a}_k - \hat{a}_k, \quad 1 \leq k \leq K, \quad (7.70)$$

then it is simple to show that the distance squared between the two signals is

$$d^2 = \sum_{i=1}^K \sum_{j=1}^K \epsilon_i \epsilon_j^* \rho_h(j-i). \quad (7.71)$$

Substituting from (7.59), this distance can be expressed as a discrete-time distance

$$d^2 = A_h^2 \cdot \sum_{k=1}^{\infty} \left| \sum_{i=1}^K \epsilon_i g_{h,k-i} \right|^2. \quad (7.72)$$

The minimum distance  $d_{\min}^2$  is the minimization of (7.72) over all non-zero difference sequences  $\{\epsilon_k, 1 \leq k \leq K\}$ . This minimization problem will be considered further in Chapter 8.

## 7.4. BANDWIDTH and SIGNAL DIMENSIONALITY

This chapter has introduced the important concept of the dimensionality of the subspace spanned by the set of known signals. In this section, we develop additional insight into the relationship between this dimensionality and the bandwidth required to accommodate this set of signals.

One important property of a signal design is the *spectral efficiency* (defined in (6.7)). Chapter 6 also derived a generalized Nyquist criterion, which established the minimum bandwidth required to eliminate ISI at a matched filter output for  $N$  orthogonal pulses. A set of pulse waveforms  $h_n(t)$ ,  $1 \leq n \leq N$  satisfies the generalized Nyquist criterion if

$$\int_{-\infty}^{\infty} h_n(t) h_m^*(t - kT) dt = h_n(t) * h_m^*(-t) \Big|_{t=kT} = \sigma_h^2 \delta_k \delta_{m-n}, \quad (7.73)$$

or in words  $h_n(t)$  is orthogonal to its own translates by multiples of the symbol interval  $T$  and also to all translates of  $h_m(t)$  for  $m \neq n$ . When (7.73) is satisfied, we can successfully separate out the orthogonal signals using a bank of matched filters at the receiver, and we can also guarantee no ISI. It was shown in Chapter 6 that (7.73) can be satisfied if and only if the collective bandwidth of the pulses is at least  $N \cdot \pi/T$  radians.

In this section, we will understand this result better, by examining it from a different perspective. In Section 7.1 we defined the subspace of signals spanned by a set of  $L$  known signals, and observed that this subspace is finite dimensional (with dimension  $N$ ). Our goal is to understand better the relationship between the bandwidth of the signals and the dimension of the subspace.

### 7.4.1. Landau-Pollak Theorem

No signal can be both time limited and bandlimited. A bandlimited signal is not time limited, in the sense that its energy cannot be totally confined to any finite interval of time, and a time-limited function is not bandlimited, in the sense that its energy cannot be totally confined to a finite band of frequencies. However, it is possible for functions to be bandlimited and *approximately* time limited, or time limited and *approximately* bandlimited. For example, consider a function  $f(t)$  that is causal and bandlimited to  $B$  Hz, and also has finite energy  $\sigma_f^2$ . Then  $f(t)$  never goes precisely to zero beyond any fixed time  $t_0$ , but because it has finite energy it will decay gradually to zero. One way to measure the rate of that decay is to calculate the fraction  $\epsilon(t_0)$  of its energy outside the interval  $[0, t_0]$ , where  $\epsilon(t_0) < 1$ . Specifically, let

$$\int_0^{t_0} |f(t)|^2 dt = \sigma_f^2 \cdot (1 - \epsilon(t_0)). \quad (7.74)$$

Since a fraction  $\epsilon(t_0)$  of the energy is outside the interval, a fraction  $1 - \epsilon(t_0)$  is within the interval. For a causal finite-energy function  $f(t)$ , as  $t_0 \rightarrow \infty$ ,  $\epsilon(t_0) \rightarrow 0$ . If we define the signal to be approximately time limited to an interval when less than a specific fraction  $\epsilon$  of its energy is outside that interval, then we can always choose a

large enough interval that the signal is approximately time limited.

Although the signal space of all finite-energy signals is infinite-dimensional, it is also true that the subset of such signals that are bandlimited to  $B$  Hz and approximately time limited to  $[0, t_0]$  is approximately finite dimensional, with dimension  $2Bt_0 + 1$ . This statement is made rigorous by the *Landau-Pollak theorem* [1].

#### Theorem.

There exists a set of  $2Bt_0 + 1$  orthonormal waveforms  $\phi_i(t)$ , such that for any (possibly complex-valued) finite-energy waveform  $f(t)$  with energy  $\sigma_f^2$  that is bandlimited to  $B$  Hz, for any constant  $0 < \epsilon < 1$ , and for any  $t_0$  sufficiently large that

$$\int_0^{t_0} |f(t)|^2 dt > \sigma_f^2 (1 - \epsilon), \quad (7.75)$$

there exists a set of  $2Bt_0 + 1$  coefficients  $f_i$  such that

$$\int_{-\infty}^{\infty} |f(t) - \sum_{i=0}^{2Bt_0} f_i \phi_i(t)|^2 dt < 12 \sigma_f^2 \epsilon. \quad (7.76)$$

□

We can state this theorem in words as follows. If less than a fraction  $\epsilon$  of a bandlimited signal's energy is outside an interval  $[0, t_0]$ , then that signal can be approximated by a linear combination of a set of  $2Bt_0 + 1$  orthonormal waveforms with an error which has energy less than a fraction  $12\epsilon$  of the signal's energy. Thus, the dimensionality of the subspace of all signals approximately time limited to  $t_0$  and bandlimited to  $B$  is approximately  $2Bt_0 + 1$ , in the sense that a small fraction of the signal's energy is outside a signal subspace of dimension  $2Bt_0 + 1$ . As  $t_0$  increases, the fraction of energy outside this subspace (which is also growing in dimensionality) gets smaller.

### 7.4.2. Relation to the Generalized Nyquist Criterion

In the generalized Nyquist criterion, we made no attempt to time-limit the pulse waveforms  $h_n(t)$  to the symbol interval  $T$ . Thus, the Landau-Pollak theorem does not apply directly. However, the generalized Nyquist criterion and the Landau-Pollak theorem are connected, and consistent with one another, as we now show.

The key to forming the connection is to consider a sequence of  $K$  transmitted symbols. Suppose  $h_n(t)$ ,  $1 \leq n \leq N$  is a set of pulses bandlimited to  $B$  Hz that satisfy the generalized Nyquist criterion. Generate a PAM plus orthogonal multipulse signal consisting of  $K$  symbols,

$$s(t) = \sum_{k=0}^{K-1} \sum_{n=1}^N A_{k,n} h_n(t - kT). \quad (7.77)$$

Since  $s(t)$  is a linear combination of  $NK$  orthogonal waveforms  $h_n(t - kT)$ ,  $1 \leq n \leq N$ ,  $0 \leq k \leq K-1$ , it lies in an  $NK$ -dimensional subspace of signal space. It is also easy to show (see Problem 7-12) that under very mild conditions,  $s(t)$  is approximately time limited to  $[0, KT]$  in the sense that the fraction of the energy of

$s(t)$  outside this interval goes to zero as  $K \rightarrow \infty$ . Thus, the Landau-Pollak theorem tells us that  $s(t)$  can be approximated by  $2BKT + 1$  orthonormal functions, with increasing accuracy as  $K \rightarrow \infty$ . This means that this dimensionality must be at least the actual dimensionality  $NK$ ,

$$2BKT + 1 \geq NK, \quad B \geq \frac{NK - 1}{2KT}. \quad (7.78)$$

As  $K \rightarrow \infty$ , the Landau-Pollak theorem implies that the bandwidth required is  $B \geq N/2T$  Hz. Since this equals  $N \cdot \pi/T$  radians/sec, this is consistent with the generalized Nyquist criterion.

### 7.4.3. Impact of Signal Bandwidth on the Isolated Pulse

One impact of the Landau-Pollak theorem is that the parameter  $2Bt_0$ , the so-called *time-bandwidth product*, plays an important role for signals that are approximately time limited and bandlimited. For a bandlimited signal with bandwidth  $B$ , as  $2Bt_0$  increases, a couple of things happen:

- The fraction of the signal energy confined to an appropriate time interval of duration  $t_0$  will increase.
- The fraction of the signal energy falling outside a  $2Bt_0 + 1$  dimensional subspace of signal space will decrease.

When  $2Bt_0$  is small, the notion of a pulse being confined to an interval of duration  $t_0$  is crude at best. However, as  $2Bt_0$  gets large, we can design bandlimited pulses that are, for all practical purposes, confined to the interval of duration  $t_0$ .

The Landau-Pollak theorem considers a waveform with bandwidth  $B$  and requires us to find a sufficiently large time limit  $t_0$  such that most of the energy of the waveform lies within  $[0, t_0]$ . An alternative approach is to hold  $t_0$  fixed and increase the bandwidth  $B$ , allowing the waveform to be increasingly confined to  $[0, t_0]$ . The dual notions of increasing the bandwidth or the time interval both arise in digital communication.

#### Example 7-15.

In spread spectrum, a single pulse  $h(t)$  is amplitude modulated for each data symbol. The Nyquist criterion says that a bandwidth of  $\pi/T$  is required if ISI is to be avoided. In fact, in spread spectrum the bandwidth  $B$  is much larger (often hundreds of times), so that  $2BT$  is very large. In this case, it is possible to make the pulse  $h(t)$  nearly time limited to the symbol interval  $T$ . This implies in turn that ISI is rarely a practical issue in spread spectrum systems. In fact, countering or avoiding ISI is often a motivation for using spread spectrum; the essential property is that the time-bandwidth product is very large. This issue is addressed further in Chapter 8.  $\square$

#### Example 7-16.

In orthogonal multipulse modulation (for example FSK), one of  $N$  orthogonal pulses is transmitted for each symbol. A side effect is that the minimum bandwidth required to satisfy the generalized Nyquist criterion is  $N \cdot \pi/T$  radians, or  $2BT = N$  where  $B$  is in Hz. If  $N$  is large (not two or three), a side effect of this larger bandwidth is that the pulses can be

designed to be largely confined to the symbol interval  $T$ . This is not a primary motivation for using orthogonal multipulse, but rather a side effect. One advantage of orthogonal multipulse can thus be less susceptibility to ISI.  $\square$

**Example 7-17.**

In multicarrier modulation, the usual perspective is that as the dimensionality of the signal set is increased, the bandwidth  $B$  is kept constant but the symbol interval  $T$  is increased. While the symbol rate is thereby reduced, a number of orthogonal pulses, each independently amplitude modulated with their own data symbols, can be transmitted simultaneously and separated out at the receiver by a bank of matched filters, keeping the spectral efficiency approximately constant. In fact, a maximum of  $N = 2BT$  orthogonal pulses can be defined consistent with generalized Nyquist criterion, and with this maximum  $N$ , the spectral efficiency is not affected by increasing  $N$  and  $T$ . One side effect is that the pulses can be designed to be more confined to a symbol interval, in this case because of the increase in  $T$  for constant  $B$ . Again, the system can be less susceptible to ISI as a result. Often a short guard time between pulses will completely eliminate ISI.  $\square$

## 7.5. FURTHER READING

The approach that has been followed in this chapter, that of considering receiver design in the absence of noise considerations, is not standard in textbooks or the literature. The benefit of this approach is that a wealth of receiver structures have been quickly derived from a common design principle. It turns out that this principle, minimum-distance receiver design, is optimal in a certain sense (defined in Chapter 9) for channels with additive white Gaussian noise. Thus, these receiver structures are usually derived from noise considerations, an approach that is more circuitous.

Viewing receiver design from a minimum-distance perspective also gives us an understanding of the relationship between the geometric properties of the signal set and the receiver noise immunity, as measured by minimum distance. It will be shown in Chapter 8 that minimum distance is indeed a direct measure of receiver noise immunity for additive Gaussian noise, as we argued here from an intuitive perspective.

The valuable notion of viewing signals geometrically in signal space was popularized by the book of Wozencraft and Jacobs [2], which remains recommended reading.

## PROBLEMS

- 7-1. Define  $d_{i,j} = \|S_i - S_j\|$ . Show that  $Y = S_i + E$  is closer to  $S_j$  than to  $S_i$  if and only if

$$\operatorname{Re}\{ \langle E, U \rangle \} > \frac{d_{i,j}}{2} \quad (7.79)$$

where  $U$  is a unit vector in the direction of  $(S_j - S_i)$ .

7-2.

- (a) Give an example of a pulse  $h(t)$  with time-duration that is exactly two symbol periods ( $2T$ ) (and hence it is not bandlimited) and obeys the Nyquist criterion at the output of a matched filter,  $\rho_h(k) = \sigma_h^2 \delta_k$ .
- (b) Repeat a. for three symbol periods ( $3T$ ).

7-3.

- (a) Show that the pulse autocorrelation obeys the symmetry relation  $\rho_h(k) = \rho_h^*(-k)$ .
- (b) Show that the folded spectrum is non-negative real valued on the unit circle.

7-4. Define

$$S_{h,+}(z) = \sum_{k=0}^{\infty} \rho_h(k) z^{-k}, \quad (7.80)$$

and show that the folded spectrum is

$$S_h(z) = S_{h,+}(z) + S_{h,+}^*(1/z^*) - \rho_h(0). \quad (7.81)$$

This gives a convenient way to calculate the folded spectrum.

- 7-5. Generalize Example 7-11 as follows. Let  $h_0(t)$  be a complex-valued pulse shape that has energy  $\sigma_0^2$  and is orthogonal to all its translates by multiples of the symbol interval  $T$ . Let  $F(z) = \sum_{k=0}^K f_k z^{-k}$  be a general  $K$ -th order FIR filter, and define a pulse shape

$$h(t) = \sum_{k=0}^K f_k h_0(t - kT). \quad (7.82)$$

- (a) Show that

$$\rho_h(k) = \sigma_0^2 f_k * f_{-k}^*, \quad (7.83)$$

and hence that

$$S_h(z) = \sigma_0^2 F(z) F^*(1/z^*). \quad (7.84)$$

- (b) What is the pulse energy?

- 7-6. Describe the operation of the minimum-distance receiver of Figure 7-10a for the following transmitter coder strategy: Four bits of information are transmitted as three successive symbols chosen from the alphabet  $\{\pm 1, 0\}$ . Only 16 possible combinations of three successive symbols out of  $3^3 = 27$  are used. How many sequences of data symbols must the receiver consider?
- 7-7. Show that the minimum-distance receiver of Figure 7-11 reduces to that of Figure 7-10 when there is no ISI.
- 7-8. In this problem, we will derive an alternative geometrical interpretation, due to John Barry, of the receiver structure in Figure 7-11. Let  $M_h$  be the subspace of signal space spanned by the  $K$  translates of the known signal pulses,  $\{h(t - kT), 1 \leq k \leq K\}$ .
- (a) Show that for any signal  $X \in M_h$ , where  $x(t) \leftrightarrow X$ , when  $x(t)$  is input to the top arm of Figure 7-11,

$$\|X\|^2 = \int_{-\infty}^{\infty} |x(t)|^2 dt = \sum_{k=1}^{\infty} |w_k|^2. \quad (7.85)$$

Thus, for input continuous-time signals restricted to  $M_h$ , the filter is norm-preserving.

- (b) Show that for any input signal  $Y \leftrightarrow y(t)$ , and any set of vectors  $\{X_i \in M_h\}$ , minimizing  $\|Y - X_i\|^2$  over  $i$  is the same as minimizing  $\|Y_h - X_i\|^2$  over  $i$ , where  $Y_h$  is the projection of  $Y$  on  $M_h$ . Thus, since the minimum-distance receiver wishes to determine the distance between the received signal  $Y$  and a set of known signals, where the latter are all in  $M_h$ , it suffices to first determine the projection  $Y_h$ , and then calculate the distances.
- (c) Show that the discrete-time response to continuous-time input  $X = Y - Y_h$  is identically zero.
- (d) Show that if  $X_i \in M_h$  and the responses of the filter to  $Y$  and  $X_i$  are  $w_{y,k}$  and  $w_{i,k}$ , then

$$\|Y - X_i\|^2 = \|Y - Y_h\|^2 + \sum_{k=0}^{\infty} |w_{y,k} - w_{i,k}|^2. \quad (7.86)$$

This basic result proves that minimizing the discrete-time distance over  $i$  is equivalent to minimizing the continuous-time distance.

- (e) Using the result of (d), interpret the minimum-distance receiver structure of Figure 7-11.
- 7-9. Suppose we add a first-order rational causal allpass filter before the Euclidean norm in Figure 7-12, or equivalently in both the forward paths. Combine the allpass filter with the precursor equalizer in the top path, and combine the allpass filter with the channel model filter in the bottom path. For simplicity, assume  $A_h^2 = 1$ .
- (a) Determine the resulting transfer functions of the precursor equalizer and discrete-time channel model filters.
- (b) Show that it is not possible to use the allpass filter to turn the new precursor equalizer into a causal filter.
- (c) Determine the impulse response of the new channel model filter.
- 7-10. Suppose we add a first-order rational causal allpass filter before the Euclidean norm in Figure 7-13, or equivalently in both the forward paths. As in Problem 7-9, combine the allpass filter with the precursor equalizer and channel-model filter.
- (a) Determine the transfer functions of the precursor equalizer and discrete-time channel model filters.
- (b) Show that it is possible to turn the new precursor equalizer filter in the upper path into a causal filter by choosing the allpass filter appropriately. What is the allpass filter, and what are the resulting transfer functions of the precursor equalizer and channel-model filters?
- (c) Determine the impulse response of the new channel model filter for the general case of a.
- (d) Repeat c. for the particular allpass filter of b.
- 7-11. Calculate the minimum distance  $d_{\min}$  for the following cases:
- (a) Slicer design, with the data-symbol alphabet 4-PSK and magnitude unity.
- (b) Isolated pulse PAM, where the pulse energy is  $\sigma_h$  and the data symbol is chosen from the same alphabet as in a.
- (c) Orthogonal multipulse with pulse energy  $\sigma_h^2$ .
- (d) PAM with ISI for two data symbols ( $K = 2$ ) and a pulse autocorrelation function  $\rho_h(k) = \alpha^{|k|}$  for a real-valued  $0 \leq \alpha < 1$ .
- 7-12. Assume that in (7.77), the causal orthogonal bandlimited pulses  $h_n(t)$  are chosen such that their tail energy is bounded by



$$\int_{t_0}^{\infty} h_s^2(t) dt \leq \frac{\alpha}{t_0^2}, \quad (7.87)$$

for some constant  $\alpha$ . Thus, the energy falls off at least as the square of time. Show that the fraction of the energy in  $s(t)$  falling outside the interval  $[0, KT]$  goes to zero as  $K \rightarrow \infty$ . Thus, the signal becomes approximately time limited to  $[0, KT]$  as  $K \rightarrow \infty$ .

## REFERENCES

1. H.J.Landau and H.O.Pollak, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty. III The Dimension of the Space of Essentially Time- and Band-Limited Signals," *Bell Sys. Tech. Journal* **41** p. 1295 (1962).
2. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York (1965).

# 8

---

## NOISE

---

In Chapter 7 we developed a systematic technique for designing receivers. In this chapter, we analyze the probability of error performance of these receivers for a channel model that includes additive white Gaussian noise. As discussed in Chapter 5, Gaussian noise models thermal noise and high-density shot noise, as found in cable, radio, and fiber optic channels.

In Section 8.8, the design of fiber optic receivers is considered for the regime where quantum noise, and not Gaussian noise, is the dominant impairment. In particular, we will derive a fundamental limit on the probability of error, known as the quantum limit, and then display receiver structures that can actually achieve this limit.

In this chapter, we analyze the noise performance assuming the minimum-distance design criterion of Chapter 7, but still do not establish the optimality of that receiver design methodology. Optimality is considered in Chapter 9 for stationary Gaussian noise on the channel and for a particular definition of optimality.

### 8.1. COMPLEX-VALUED GAUSSIAN PROCESSES

In Section 3.2, the real-valued Gaussian random process was defined as a process for which any arbitrary set of samples is jointly Gaussian. A *complex-valued Gaussian random process* consists of two jointly Gaussian real-valued processes, a real part and an imaginary part. By jointly Gaussian, we mean that any arbitrary set of samples

of the real and imaginary parts is a jointly Gaussian set of random variables. Such processes are important in digital communications in modeling the baseband-equivalent noise. Complex-valued processes have some special properties that distinguish them from real-valued processes. In this section, we will study those properties as a preliminary to characterizing the noise performance of the modulation systems.

Let  $Z(t)$  be a zero-mean complex-valued Gaussian process. Since  $Z(t)$  is complex-valued, it consists of two real-valued processes,

$$R(t) = \text{Re}\{Z(t)\}, \quad I(t) = \text{Im}\{Z(t)\}. \quad (8.1)$$

By assumption both  $R(t)$  and  $I(t)$  are zero-mean Gaussian processes. To fully characterize the statistics of  $Z(t)$ , we must specify the *joint* statistics of  $R(t)$  and  $I(t)$ . Because they are Gaussian and zero-mean,  $R(t)$  and  $I(t)$  are fully characterized by their second order statistics,

$$\begin{aligned} R_R(\tau) &= E[R(t+\tau)R(t)], & R_I(\tau) &= E[I(t+\tau)I(t)], \\ R_{RI}(\tau) &= E[R(t+\tau)I(t)]. \end{aligned} \quad (8.2)$$

The complex-valued process  $Z(t)$  is strictly stationary if  $R(t)$  and  $I(t)$  are jointly wide-sense stationary, and hence jointly strictly stationary (since they are Gaussian).  $R(t)$  and  $I(t)$  are jointly wide-sense stationary if the correlation functions  $E[R(t+\tau)R(t)]$ ,  $E[I(t+\tau)I(t)]$ , and  $E[R(t+\tau)I(t)]$  are not functions of  $t$ , as indicated in (8.2).

According to the definition, the *complex* process  $Z(t)$  is wide-sense stationary if the autocorrelation function

$$R_Z(\tau) = E[Z(t+\tau)Z^*(t)], \quad (8.3)$$

is not a function of  $t$ . This is not the same as saying that the real and imaginary parts are jointly wide-sense stationary. Clearly,  $R_Z(\tau)$  could not by itself contain information equivalent to  $R_R(\tau)$ ,  $R_I(\tau)$ , and  $R_{RI}(\tau)$ , since (8.2) constitutes *three* functions of  $\tau$ , while  $R_Z(\tau)$  specifies only *two* functions of  $\tau$ , its real and imaginary parts. Thus, we require more than  $R_Z(\tau)$  to fully specify the statistics of  $Z(t)$ . In addition to  $R_Z(\tau)$ , it suffices to know the *complementary autocorrelation function*, defined as

$$\tilde{R}_Z(\tau) = E[Z(t+\tau)Z(t)]. \quad (8.4)$$

Again, since  $\tilde{R}_Z(\tau)$  can be expressed in terms of  $R_R(\tau)$ ,  $R_I(\tau)$ , and  $R_{RI}(\tau)$ , it must not be a function of  $t$  if  $Z(t)$  is to be strict-sense stationary.

Using the relations  $2 \cdot R(t) = Z(t) + Z^*(t)$  and  $2j \cdot I(t) = Z(t) - Z^*(t)$ , it is easy to show that

$$\begin{aligned} 2 \cdot R_R(\tau) &= \text{Re}\{R_Z(\tau)\} + \text{Re}\{\tilde{R}_Z(\tau)\}, & 2 \cdot R_I(\tau) &= \text{Re}\{R_Z(\tau)\} - \text{Re}\{\tilde{R}_Z(\tau)\}, \\ 2 \cdot R_{RI}(\tau) &= \text{Im}\{\tilde{R}_Z(\tau)\} - \text{Im}\{R_Z(\tau)\}. \end{aligned} \quad (8.5)$$

For the special case of a real-valued  $Z(t)$ ,  $R_I(\tau) = R_{RI}(\tau) = 0$ , and  $R_Z(\tau) = \tilde{R}_Z(\tau)$ . Given *both* the autocorrelation and complementary autocorrelation functions, (8.5) allows us to determine the full complement of joint statistics of  $R(t)$  and  $I(t)$ . Conversely, neither the autocorrelation nor the complementary autocorrelation

function is sufficient by itself to fully specify the statistics of  $Z(t)$ .

$Z(t)$  is wide-sense stationary if it has an autocorrelation function  $E[Z(t + \tau)Z^*(t)] = R_Z(\tau)$  that is independent of  $t$ . In that case, its power spectrum  $S_Z(j\omega)$  is the Fourier transform of  $R_Z(\tau)$ . However, in the case of complex Gaussian processes, wide-sense stationarity does *not* imply strict sense stationarity, because even for a wide-sense stationary process  $E[Z(t + \tau)Z(t)]$  may be a function of  $t$ . However, (8.5) implies that if a Gaussian process is wide-sense stationary, and in addition  $E[Z(t + \tau)Z(t)]$  is not a function of  $t$ , then the real and imaginary parts are jointly wide-sense stationary, and  $Z(t)$  is strictly stationary.

Based on these considerations, there are two important differences between real-valued and complex-valued Gaussian processes:

- A complex-valued zero-mean Gaussian process is fully specified by both the autocorrelation and complementary autocorrelation functions, but not by either one alone. In particular, it is not fully characterized by its power spectrum. In contrast, a real-valued zero-mean Gaussian process requires only the autocorrelation function, and is thus fully characterized by its power spectrum.
- A wide-sense stationary complex-valued zero-mean Gaussian process is not necessarily strictly stationary, but it is strictly stationary if *both* the autocorrelation and complementary autocorrelation functions are not functions of  $t$ . In contrast, a real-valued zero-mean Gaussian process is strictly stationary if and only if it is wide-sense stationary.

Although there are significant differences between real-valued and complex-valued Gaussian processes, there is an important special case, considered next, where the two have similar properties.

### 8.1.1. Circularly Symmetric Gaussian Processes

Let a single random variable  $Z = (R + jI)$  be complex-valued, Gaussian, and zero-mean. Then, calculating

$$E[Z^2] = E[R^2] - E[I^2] + 2jE[RI], \quad (8.6)$$

we notice that  $R$  and  $I$  are identically distributed (have the same variance) and independent ( $E[RI] = 0$ ) if and only if  $E[Z^2] = 0$ . A complex-valued Gaussian random variable will be called *circularly symmetric* if  $E[Z^2] = 0$ . [1,2]. The source of the terminology is that the probability density function of  $Z$  is circularly symmetric, or

$$P_{R,I}(r, i) = \frac{1}{2\pi\sigma^2} e^{-(r^2 + i^2)/2\sigma^2}, \quad (8.7)$$

where  $\sigma^2$  is the variance of the real and imaginary parts.

This concept can be generalized. A complex-valued zero-mean Gaussian process is circularly symmetric if

$$E[Z(t + \tau)Z(t)] = 0, \quad \text{for all } t \text{ and } \tau. \quad (8.8)$$

Such processes have a number of important simplifying properties, and further, most

Gaussian processes encountered in digital communication are circularly symmetric. Note that a real-valued process *cannot* be circularly symmetric since for such a process  $R_Z(\tau) = \tilde{R}_Z(\tau)$ .

First of all, a circularly symmetric Gaussian process is strictly stationary if and only if it is wide-sense stationary, since then the real and imaginary parts are jointly wide-sense stationary. For a wide-sense stationary circularly symmetric Gaussian process, (8.5) simplifies to

$$\begin{aligned} 2 \cdot R_R(\tau) &= \text{Re}\{R_Z(\tau)\}, \quad 2 \cdot R_I(\tau) = \text{Re}\{R_Z(\tau)\}, \\ 2 \cdot R_{RI}(\tau) &= -\text{Im}\{R_Z(\tau)\}. \end{aligned} \quad (8.9)$$

Based on (8.9), circularly symmetric Gaussian processes have several nice properties:

- The real and imaginary parts individually have identical statistics, by virtue of having the same autocorrelation function  $\frac{1}{2}\text{Re}\{R_Z(\tau)\}$ .
- Since  $R_Z(0)$  must be real valued,  $\text{Im}\{R_Z(0)\} = R_{RI}(0) = 0$ . This implies that for any given time  $t$ ,  $R(t)$  and  $I(t)$  are uncorrelated and hence statistically independent, although they are not necessarily uncorrelated nor independent when sampled at different times.
- Circularly symmetric processes with a *real-valued* autocorrelation function  $R_Z(\tau)$  have a real and imaginary part that are independent at *all* times, since  $R_{RI}(\tau) = 0$ . (Note that a real-valued  $R_Z(\tau)$  does not imply that the process is real-valued. In fact, a circularly symmetric  $Z(t)$  with real-valued  $R_Z(\tau)$  *cannot* be a real-valued process!)  $R_Z(\tau)$  is real-valued when the power spectrum of the process (which is *always* real-valued) has even symmetry about  $\omega = 0$ .

Circular symmetry is preserved by linear time-invariant filtering. That is, if we apply a circularly symmetric Gaussian process  $Z(t)$  to a linear time-invariant system with impulse response  $h(t)$ , then the output is given by a convolution

$$V(t) = \int_{-\infty}^{\infty} h(\tau)Z(t - \tau) d\tau, \quad (8.10)$$

and

$$E[V(t + \tau)V(t)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(u)h(v)E[Z(t + \tau - u)Z(t - v)] du dv = 0, \quad (8.11)$$

since the integrand is identically zero. More generally, the circularly symmetric property is preserved by time-varying linear systems, such as modulators.

### Example 8-1.

Assume that  $Z(t)$  is a circularly symmetric stationary zero-mean Gaussian random process, and define  $V(t) = e^{j\omega_c t} Z(t)$ .  $V(t)$  is also stationary and circularly symmetric, since

$$R_V(\tau) = E[V(t + \tau)V^*(t)] = e^{j\omega_c \tau} R_Z(\tau), \quad (8.12)$$

$$\tilde{R}_V(\tau) = e^{j\omega_c (2t + \tau)} E[Z(t + \tau)Z(t)] = 0. \quad (8.13)$$

Further, the power spectrum of  $V(t)$  is  $S_Z(j(\omega - \omega_c))$ .  $\text{Re}\{V(t + \tau)\}$  and  $\text{Im}\{V(t)\}$  are thus independent for all  $\tau$  if and only if  $S_Z(j\omega)$  is symmetric about  $\omega_c$ ; that is,  $S_Z(j(\omega_c + \Delta\omega)) = S_Z(j(\omega_c - \Delta\omega))$  for all  $\Delta\omega$ . If  $Z(t)$  is a narrowband process, this implies that  $Z(t)$  has only positive-frequency components in its power spectrum.  $\square$

A complex-valued Gaussian process obtained from a real-valued Gaussian process by modulation is neither stationary nor circularly symmetric, as illustrated by the following example.

### Example 8-2.

Let  $N(t)$  be a real-valued zero-mean Gaussian random process. Thus,  $N(t)$  cannot be circularly symmetric, and, not surprisingly, neither is  $Z(t) = e^{j\omega_c t} N(t)$ . In particular,

$$R_Z(\tau) = E[Z(t + \tau)Z^*(t)] = e^{j\omega_c \tau} R_N(\tau), \quad (8.14)$$

$$E[Z(t + \tau)Z(t)] = e^{j\omega_c 2t + \tau} R_N(\tau). \quad (8.15)$$

$Z(t)$  is wide-sense stationary, but it is neither strictly stationary nor circularly symmetric. That  $Z(t)$  is non-stationary is not surprising, since at certain times (the zero-crossings of the carrier) the real part of  $Z(t)$  is identically zero, and similarly for the imaginary part. That  $Z(t)$  is wide-sense stationary in spite of not being strictly stationary is perhaps surprising to those accustomed to real-valued Gaussian processes.  $\square$

## Discrete-Time Gaussian Processes

All the properties we have described carry over to discrete-time zero-mean Gaussian random processes. In particular, such a complex-valued process  $Z_k$  is fully characterized by  $R_Z(m)$  and  $\bar{R}_Z(m)$ . By definition, it is circularly symmetric if

$$E[Z_{k+m}Z_k] = 0, \quad \text{for all } m \text{ and } k. \quad (8.16)$$

If  $Z_k = Z(kT)$  is obtained by sampling a circularly symmetric continuous-time process, it will itself be circularly symmetric. If  $Z_k$  is circularly symmetric, its real and imaginary parts have the same variance, and are independent at a given time  $t$ . Further, the real and imaginary parts are statistically independent for all time if and only if the autocorrelation  $R_Z(k)$  is real-valued. As in continuous time, circular symmetry is preserved by linear time-invariant filtering, and more generally by linear operations.

## White Gaussian Processes

An important subclass of zero-mean complex Gaussian processes are *white*. Such processes have an autocorrelation function

$$R_Z(\tau) = N_0 \delta(\tau), \quad R_Z(k) = 2\sigma^2 \delta_k, \quad (8.17)$$

for continuous and discrete time respectively. The convention is that  $\sigma^2$  is the variance of the real part or the imaginary part, so that  $2\sigma^2$  is the variance of the complex process.

For real-valued processes, in continuous time the white property implies that  $Z(t + \tau)$  and  $Z(t)$  are uncorrelated and hence independent for all  $\tau \neq 0$ , and for

discrete time,  $Z(k+m)$  and  $Z(k)$  are uncorrelated and hence independent for all  $m \neq 0$ .

A white complex-valued Gaussian process is not necessarily strict-sense stationary. However, if the process is both *white* and *circularly symmetric*, then the following properties hold:

- The real and imaginary parts of the process are identically distributed, and are each white real-valued Gaussian processes.
- The real and imaginary parts are independent of one another, since the autocorrelation function is real-valued.

Thus, circularly symmetric zero-mean white complex Gaussian processes are maximally random, in the sense that (a) the samples of the process are mutually independent and (b) the real and imaginary parts are independent.

Another important observation is that any Gaussian process obtained by a time-invariant linear filtering of a circularly symmetric white Gaussian process is itself circularly symmetric, although in general it will not be white (unless the filter is allpass).

## 8.2. FUNDAMENTAL RESULTS

We will now calculate the probability of error for a particular  $N$ -dimensional complex Euclidean formulation of the minimum-distance receiver design for a set of known signals in Gaussian noise. It will turn out that this formulation is general enough to cover all the cases of interest in the remainder of this chapter.

### Gaussian Noise Vectors

Let  $\mathbf{Z}' = [Z_1, Z_2, \dots, Z_N]$  be a complex-valued zero-mean Gaussian vector (where  $\mathbf{Z}'$  denotes the matrix transpose of  $\mathbf{Z}$ ). In the sequel we will assume that  $\mathbf{Z}$  has several simplifying properties:

- The components of  $\mathbf{Z}$  are uncorrelated, that is,  $E[Z_i Z_j^*] = 0$  for  $i \neq j$ .
- The components of  $\mathbf{Z}$  are circularly symmetric (Section 8.1), or  $E[Z_i Z_j] = 0$  for  $1 \leq i, j \leq N$ . This plus the uncorrelated property implies that the components of  $\mathbf{Z}$  are mutually independent, and further that the real and imaginary parts of each component are independent.
- The components of  $\mathbf{Z}$  are identically distributed, that is  $E[|Z_n|^2] = 2\sigma^2$  for  $1 \leq n \leq N$ .

We also need to consider the real-valued  $\mathbf{Z}$  case. In this case,  $\mathbf{Z}$  cannot be circularly symmetric, but the uncorrelated assumption by itself implies that the components of  $\mathbf{Z}$  are independent. The identically distributed assumption implies that the real and imaginary components of  $\mathbf{Z}$  have the same variance  $\sigma^2$ .

Let a complex random variable  $C$  be defined as

$$C = \langle \mathbf{Z}, \mathbf{e} \rangle = \mathbf{Z}' \mathbf{e}^* \quad (8.18)$$

where  $\mathbf{e}$  is a unit-magnitude vector, i.e.  $\|\mathbf{e}\| = 1$ . Then clearly  $C$  is Gaussian, since it is a linear combination of Gaussian random variables. Further, it is circularly symmetric ( $E[C^2] = 0$ ), since it is a linear function of a circularly symmetric Gaussian vector. This implies that  $\text{Re}\{C\}$  and  $\text{Im}\{C\}$  are identically distributed and independent. To determine the statistics of  $C$ , all we have to determine is its variance. Calculating this directly,

$$\begin{aligned} E[|C|^2] &= E\left[\sum_{i=1}^N \sum_{k=1}^N Z_i Z_k^* e_i e_k^*\right] \\ &= \sum_{i=1}^N E[|Z_i|^2] |e_i|^2 = 2\sigma^2 \sum_{i=1}^N |e_i|^2 = 2\sigma^2. \end{aligned} \quad (8.19)$$

Thus  $C$  has the same variance as the components of  $\mathbf{Z}$ ,  $2\sigma^2$ , and as a result, the real and imaginary parts of  $C$  each have variance  $\sigma^2$ . This result can also be explained intuitively, since  $\langle \mathbf{Z}, \mathbf{e} \rangle$  is the projection of  $\mathbf{Z}$  on the span of a unit-magnitude vector  $\mathbf{e}$ , or the component of  $\mathbf{Z}$  in the direction of unit-vector  $\mathbf{e}$ . Since  $\mathbf{Z}$  has the same variance in each of its components, it stands to reason that the variance of the component of  $\mathbf{Z}$  in *any* direction has the same variance, not just in the direction of the principal axes.

### Vector-Valued Signal in Vector-Valued Noise

Consider a received signal that is an  $N$ -dimensional complex vector, consisting of a known signal vector and an additive complex Gaussian noise vector,

$$\mathbf{Y} = \mathbf{S}_m + \mathbf{Z}, \quad (8.20)$$

where  $\mathbf{Y}' = [Y_1, Y_2, \dots, Y_N]$  is the received signal, and  $\mathbf{S}_m' = [S_{m,1}, S_{m,1}, \dots, S_{m,N}]$  is drawn from a set of  $L$  known signals  $\{\mathbf{S}_l, 1 \leq l \leq L\}$ .

### Probability of Error

Now suppose we apply the receiver design strategy of Chapter 7 to the received signal of (8.20). That is, we choose the signal that satisfies

$$\min_l \|\mathbf{Y} - \mathbf{S}_l\|^2 \quad (8.21)$$

What is the probability of error?

We will first determine the probability that the received signal  $\mathbf{Y} = \mathbf{S}_m + \mathbf{Z}$  is closer to  $\mathbf{S}_i$  than it is to  $\mathbf{S}_m$  for  $i \neq m$ , or the probability of the event

$$\|\mathbf{Y} - \mathbf{S}_i\|^2 < \|\mathbf{Y} - \mathbf{S}_m\|^2. \quad (8.22)$$

Substituting for  $\mathbf{Y}$ ,

$$\|\mathbf{Z} - (\mathbf{S}_i - \mathbf{S}_m)\|^2 < \|\mathbf{Z}\|^2, \quad (8.23)$$

or



$$\|Z\|^2 + \|S_i - S_m\|^2 - 2 \operatorname{Re}\{ \langle Z, S_i - S_m \rangle \} < \|Z\|^2. \quad (8.24)$$

Cancelling the  $\|Z\|^2$  term and dividing both sides by

$$d_{m,i} = \|S_i - S_m\|, \quad (8.25)$$

the distance between  $S_m$  and  $S_i$ , we get equivalently

$$\operatorname{Re}\{ \langle Z, \frac{S_i - S_m}{d_{m,i}} \rangle \} > \frac{d_{m,i}}{2}. \quad (8.26)$$

The probability of event (8.26) is easily calculated, since the vector  $(S_i - S_m)/d_{m,i}$  is a unit-length vector, and hence from (8.19) the left side of (8.26) is a Gaussian random variable with variance  $\sigma^2$ . The probability of (8.26) is therefore

$$\Pr[ Y \text{ closer to } S_i \text{ than } S_m \mid Y = S_m + Z ] = Q\left(\frac{d_{m,i}}{2\sigma}\right), \quad (8.27)$$

where  $Q(\cdot)$  is the integral of the tail of the unit-variance Gaussian distribution, as defined in Chapter 3.

Using this result, we can determine the error probability for the case of two signals ( $L = 2$ ).

#### Example 8-3.

If  $L = 2$ , and if  $Y = S_1 + Z$ , then an error occurs if  $Y$  is closer to  $S_2$  than to  $S_1$ . This occurs with probability

$$\Pr[ S_2 \text{ chosen} \mid Y = S_1 + Z ] = Q\left(\frac{d_{1,2}}{2\sigma}\right). \quad (8.28)$$

□

### Bounds on the Probability of Error

The exact error probability for three or more signals can be difficult to calculate, since the minimum-distance decision boundary can be very complicated. We will illustrate special cases in Sections 8.3 and 8.4 where it is not too difficult. More generally, however, we can establish bounds on the error probability that are easy to apply. These bounds become tight as  $\sigma \rightarrow 0$ , and thus represent not only bounds, but also accurate approximations for small  $\sigma$  (small error probability). Since most digital communication systems operate at low error probability, these bounds are very useful.

The upper bound will be based on the *union bound* described in Section 3.1. For  $N$  events  $\{E_n, 1 \leq n \leq N\}$ , the union bound is

$$\Pr\left[\bigcup_{n=1}^N E_n\right] \leq \sum_{n=1}^N \Pr[E_n]. \quad (8.29)$$

Returning to the probability of error for the minimum-distance receiver design, suppose that  $S_1$  is transmitted. We are interested in the probability that one of the other signals  $S_l$ ,  $2 \leq l \leq L$  is closer to the received signal  $Y$  than  $S_1$ . If we define  $E_l$  as the event that  $S_l$  is closer to  $Y$  than  $S_1$ , then

$$\Pr[S_1 \text{ not closest to } \mathbf{Y} \mid \mathbf{Y} = \mathbf{S}_1 + \mathbf{Z}] = \Pr\left[\bigcup_{l=2}^L E_l\right] \leq \sum_{l=2}^L \Pr[E_l]. \quad (8.30)$$

Since

$$\Pr[E_l] = \Pr[\mathbf{Y} \text{ closer to } S_l \text{ than to } S_1 \mid \mathbf{Y} = \mathbf{S}_1 + \mathbf{Z}] = Q\left(\frac{d_{1,l}}{2\sigma}\right), \quad (8.31)$$

we get

$$\Pr[S_1 \text{ not closest to } \mathbf{Y} \mid \mathbf{Y} = \mathbf{S}_1 + \mathbf{Z}] \leq \sum_{l=2}^L Q\left(\frac{d_{1,l}}{2\sigma}\right). \quad (8.32)$$

This is an upper bound on the probability of error, conditional on  $S_1$  being transmitted.

It was shown in Chapter 3 that  $Q(\cdot)$  is a very steep function of its argument for large arguments (corresponding to high SNR). This implies that the sum in (8.32) tends to be dominated by the term corresponding to the smallest argument. Define  $d_{1,\min}$  as the smallest  $d_{1,l}$ ,  $2 \leq l \leq L$ . Then the union bound of (8.32) can be approximated by

$$\Pr[\mathbf{Y} \text{ not closest to } S_1 \mid \mathbf{Y} = \mathbf{S}_1 + \mathbf{Z}] \approx K_1 Q\left(\frac{d_{1,\min}}{2\sigma}\right), \quad (8.33)$$

where  $K_1$  is the number of signals that are distance  $d_{1,\min}$  away from  $S_1$ . We can no longer assert that (8.33) is an upper bound, since we have thrown away positive terms, making the right side smaller. However, for small  $\sigma$ , (8.33) remains an accurate approximation. It is also intuitive that the error probability would be dominated by the signals that are closest to  $S_1$ , since the nearest signals are the ones most likely to be confused with  $S_1$ .

A lower bound on error probability can also be established. Since  $\bigcup_{l=2}^L E_l$  contains  $E_m$  for any  $2 \leq m \leq L$ , we get that

$$\Pr\left[\bigcup_{l=2}^L E_l\right] \geq \Pr[E_m] = Q\left(\frac{d_{1,m}}{2\sigma}\right). \quad (8.34)$$

Obviously, the bound is tightest when  $d_{1,m} = d_{1,\min}$ , since that will maximize the right side of (8.34). Thus, a lower bound is

$$\Pr[S_1 \text{ not closest to } \mathbf{Y} \mid \mathbf{Y} = \mathbf{S}_1 + \mathbf{Z}] \geq Q\left(\frac{d_{1,\min}}{2\sigma}\right). \quad (8.35)$$

Together (8.33) and (8.35) establish, for small  $\sigma$ , an approximation to the error probability if  $S_1$  is transmitted that is accurate within a factor of  $K_1$ . This bound applies equally well for any other transmitted signal  $S_m$  with  $K_1$  replaced by  $K_m$  and  $d_{1,\min}$  replaced by  $d_{m,\min}$ , where  $d_{m,\min}$  is the minimum distance from  $S_m$  to any other signal, and  $K_m$  is the number of signals at distance  $d_{m,\min}$ .

We are often interested in the overall probability of error  $P_e$ , defined as the probability that the wrong signal is chosen by the minimum-distance criterion. To

calculate  $P_e$ , we must know  $\{p_l, 1 \leq l \leq L\}$ , the set of probabilities of the  $L$  signals being transmitted. Then

$$P_e = \sum_{m=1}^L \Pr[S_m \text{ not chosen} \mid \mathbf{Y} = \mathbf{S}_m + \mathbf{Z}] \cdot p_m \quad (8.36)$$

Substituting the union-bound approximation,  $P_e$  can be approximated as

$$P_e \approx \sum_{m=1}^L p_m K_m \cdot Q\left(\frac{d_{m,\min}}{2\sigma}\right). \quad (8.37)$$

As before, (8.37) will be dominated by the terms with the smallest argument to  $Q(\cdot)$ . Thus,

$$P_e \approx K \cdot Q\left(\frac{d_{\min}}{2\sigma}\right), \quad (8.38)$$

where  $K$  is a constant, called the *error coefficient*, and  $d_{\min}$  is the minimum distance between any pair of signals. The error coefficient has the interpretation as the average number of signals at the minimum distance. Since  $K$  has a much milder impact on  $P_e$  and the argument of  $Q(\cdot)$ , the error probability at high SNR is dominated by the minimum distance  $d_{\min}$ .

### 8.3. PERFORMANCE of PAM

In Chapter 6, we showed a general receiver structure for PAM (both baseband and passband) consisting of a demodulator (passband only), receive filter, sampler at the symbol rate, and slicer. The receive filter was chosen so that there was no ISI at the slicer input; that is, the pulse shape at the receive filter output satisfied the Nyquist criterion. In this section, we will characterize the performance of that receiver, as measured by the probability of error, for white Gaussian noise on the channel. The results in Section 8.2 will be directly applicable to this problem. One of the conclusions will be that there is *noise enhancement* when the channel introduces ISI that the receiver then removes, in the process increasing the noise at the slicer input.

#### 8.3.1. Equivalent Noise at Slicer

We will now characterize the statistics of the noise at the output of the receive filter when the channel adds zero-mean white Gaussian noise. In particular, let  $N(t)$  be stationary zero-mean real-valued Gaussian noise with spectrum and autocorrelation given by

$$S_N(j\omega) = N_0, \quad R_N(\tau) = N_0 \cdot \delta(\tau). \quad (8.39)$$

Assuming that this noise is additive, we can invoke linear superposition and consider the effect of the receiver front end on the signal and the noise separately. We will consider noise for a general receive filter in this section, and in Section 8.4 for the particular filters derived in Chapter 7 for the minimum-distance receiver design criterion.

Given a received signal  $Y(t)$ , the output of the receive filter in Chapter 6 is given by

$$Q(t) = (Y(t) \cdot e^{-j\omega_c t}) * \sqrt{2}f(t) \quad (8.40)$$

where  $\omega_c$  is the carrier frequency and  $f(t)$  is the impulse response of the equivalent complex-baseband receive filter. The factor of  $\sqrt{2}$  is included to ensure that the passband and baseband signals (in the absence of noise) have the same energy. The slicer input is a sampled version of  $Q(t)$ ,  $Q(kT)$ .

### Continuous-Time Receive Filter Output Noise

If the noise component of  $Y(t)$  is white Gaussian noise  $N(t)$ , and the equivalent noise at the receive filter output is complex-valued noise  $Z(t)$ , then by linearity (8.40) implies that

$$Z(t) = [N(t) \cdot e^{-j\omega_c t}] * \sqrt{2}f(t). \quad (8.41)$$

In order to determine the error probability for the slicer, we need to characterize the statistics of  $Z(t)$ .

First, since it is a linear function of a zero-mean Gaussian process,  $Z(t)$  is itself zero-mean and Gaussian. The autocorrelation function is

$$\begin{aligned} R_Z(\tau) &= E[Z(t + \tau)Z^*(t)] \\ &= E \left[ \int_{-\infty}^{\infty} N(u) \cdot \sqrt{2}e^{-j\omega_c u} f(t + \tau - u) du \int_{-\infty}^{\infty} N(v) \cdot \sqrt{2}e^{j\omega_c v} f^*(t - v) dv \right] \\ &= 2N_0 \int_{-\infty}^{\infty} f(t + \tau - u)f^*(t - u) du = 2N_0 \int_{-\infty}^{\infty} f(v)f^*(v - \tau) dv. \end{aligned} \quad (8.42)$$

Thus, complex-valued baseband noise is wide-sense stationary, and the autocorrelation and power spectrum are given by

$$R_Z(\tau) = 2N_0 f(\tau) * f^*(-\tau), \quad S_Z(j\omega) = 2N_0 |F(j\omega)|^2, \quad (8.43)$$

where the power spectrum follows because the Fourier transform of  $f^*(-\tau)$  is  $F^*(j\omega)$ . Thus,  $Z(t)$  is a wide-sense stationary complex-valued Gaussian process.

To fully characterize  $Z(t)$ , as shown in Section 8.1, we also require the complementary autocorrelation. Following the same technique,

$$\begin{aligned} E[Z(t + \tau)Z(t)] &= 2N_0 \int_{-\infty}^{\infty} e^{-j2\omega_c u} f(t + \tau - u)f(t - u) du \\ &= 2N_0 e^{-j2\omega_c t} \int_{-\infty}^{\infty} e^{j2\omega_c v} f(v + \tau)f(v) dv. \end{aligned} \quad (8.44)$$

Since  $E[Z(t + \tau)Z(t)]$  is in general a function of  $t$  as well as  $\tau$ ,  $Z(t)$  is non-stationary, in spite of the fact that it is wide-sense stationary. However, there is one special case where  $Z(t)$  is strictly stationary and circularly symmetric, namely when

$$\int_{-\infty}^{\infty} e^{j2\omega_c \nu} f(\nu + \tau) f(\nu) d\nu = 0 \quad \text{for all } \tau. \quad (8.45)$$

The left side of (8.45) is proportional to the component of  $f(t + \tau)f(t)$  at frequency  $2\omega_c$ , double the carrier frequency. For cases of practical interest, the passband signal spectrum will not overlap d.c. In this event, the bandwidth of the complex-baseband receive filter  $f(t)$  will be strictly less than the carrier frequency  $\omega_c$ . Since  $f(t + \tau)f(t)$  has bandwidth at most double the bandwidth of  $f(t)$ , it will then have bandwidth strictly less than  $2\omega_c$ , and (8.45) follows. For this special case, (8.45) implies that  $Z(t)$  satisfies

$$E[Z(t + \tau)Z(t)] = 0, \quad (8.46)$$

and is circularly symmetric.

In conclusion, for cases of practical interest the noise component  $Z(t)$  at the receive filter output is circularly symmetric and strictly stationary. Circular symmetry implies that the real and imaginary parts of  $Z(t)$  are stationary Gaussian processes with identical autocorrelation functions,

$$E[\operatorname{Re}\{Z(t + \tau)\}\operatorname{Re}\{Z(t)\}] = E[\operatorname{Im}\{Z(t + \tau)\}\operatorname{Im}\{Z(t)\}] = \frac{1}{2}\operatorname{Re}\{R_Z(\tau)\}, \quad (8.47)$$

and the cross-correlation function between the real and imaginary parts is

$$E[\operatorname{Re}\{Z(t + \tau)\}\operatorname{Im}\{Z(t)\}] = -\frac{1}{2}\operatorname{Im}\{R_Z(\tau)\}. \quad (8.48)$$

Thus, the real and imaginary parts are in general *not* independent of one another, unless  $R_Z(\tau)$  is real-valued, in which case they are independent. They are *always* independent when sampled at the same time.

### Discrete-Time Slicer Input Noise

The noise at the slicer input is  $Z_k = Z(kT)$ , a discrete-time complex-valued Gaussian random process. Since  $Z(t)$  is circularly symmetric, then so is  $Z_k$ . The real and imaginary parts of  $Z_k$  have the same variance  $\sigma^2$ , and are independent when sampled at the same time. From (8.43), the power spectrum of  $Z_k$  is

$$S_Z(e^{j\omega T}) = \frac{2N_0}{T} \sum_{m=-\infty}^{\infty} |F(j(\omega - m\frac{2\pi}{T}))|^2. \quad (8.49)$$

In general,  $\operatorname{Re}\{Z_{k+m}\}$  and  $\operatorname{Im}\{Z_k\}$  are independent for  $m \neq 0$  only if  $R_Z(k)$  is real-valued, or equivalently if  $S_Z(e^{j\omega T})$  is symmetric about  $\omega = 0$ .

### Baseband Case

The baseband case is slightly different and must be treated separately. The noise at the slicer input for this case is given by

$$Z(t) = N(t) * f(t), \quad (8.50)$$

which is real-valued. The power spectrum of this noise is

$$S_Z(j\omega T) = N_0 |F(j\omega)|^2 \quad (8.51)$$

and the power spectrum of the discrete-time sampled noise  $Z(kT)$  is

$$S_Z(e^{j\omega T}) = \frac{N_0}{T} \sum_{m=-\infty}^{\infty} |F(j(\omega + m\frac{2\pi}{T}))|^2. \quad (8.52)$$

For the same receive filter, therefore, the noise in the baseband case is half as large. One interpretation is that the passband signal requires twice the bandwidth of the baseband signal, allowing twice the noise power in the signal bandwidth. However, the passband noise is split between the real and imaginary parts, and each component of the noise is the same as in the baseband case.

### Noise Enhancement Due to the Receive Filter

In the presence of ISI, the receive filter output pulse shape typically satisfies the Nyquist criterion. Often this requires a gain in the receive filter that compensates for the channel loss, with a resulting increase in the noise power. This is called *noise enhancement*. We can see this by defining, as in Chapter 6, the pulse shape at the receiver filter output as  $p(t)$ , chosen to satisfy the Nyquist criterion. Then  $G(j\omega)B(j(\omega + \omega_c))F(j\omega) = P(j\omega)$ , where  $G(j\omega)$  is the transmit filter and  $B(j\omega)$  is the channel transfer function. The noise variance per dimension (that is, for the real or imaginary parts) at the slicer input is then

$$\sigma^2 = N_0 \cdot \int_{-\infty}^{\infty} \frac{|P(j\omega)|^2}{|G(j\omega)B(j(\omega + \omega_c))|^2} \frac{d\omega}{2\pi}. \quad (8.53)$$

At frequencies where  $B(j\omega)$  is small, and  $P(j\omega)$  is not, the integrand becomes large; that is, there is amplification of the noise on the channel due to the compensation of channel attenuation in the receive filter.

### 8.3.2. Probability of Symbol Error

Now that we have characterized the noise at the slicer input, we are prepared to determine the error probability. We will use the minimum-distance slicer design developed in Section 7.2.1, where the detected data symbol is the one closest to the complex sample at the slicer input. The results of Section 8.2, when specialized to  $K = 1$  dimensions, apply directly to this problem.

First, it is useful to summarize the results of Section 8.3.1. The slicer input  $Q_k$  is of the form

$$Q_k = A_k + Z_k. \quad (8.54)$$

where  $A_k$  is the transmitted data symbol, chosen from alphabet  $\{a_m, 1 \leq m \leq M\}$ , and  $Z_k$  is a complex-valued Gaussian noise. In the baseband case, all the quantities in (8.54) are real-valued, and the noise variance at the one-dimensional slicer input is  $\sigma^2$ , which is the power spectrum  $N_0$  of the noise times the energy in the receive filter impulse response. The passband receive filter was normalized so that the baseband and passband signal energy are the same, resulting in an extra factor of  $\sqrt{2}$  in the receive filter. The result is that the variance of the noise in the passband case

$(E[|Z_k|^2] = 2 \cdot \sigma^2)$  is double that of the baseband case. Intuitively, this is due to the fact that the bandwidth of the receive passband filter is double that of the baseband case. However, the passband data symbol and noise are complex-valued, and the variance of the noise per-dimension is the same for the baseband and passband cases ( $\sigma^2$ ). That is, real and imaginary parts of the passband noise have the same variance  $\sigma^2$ , the same as the variance of the baseband noise for the equivalent receive filter.

To calculate the probability of error, recall that, independent of the receive filter, the real and imaginary components of one sample of the complex-baseband noise are statistically independent. The only assumption required was that the passband spectrum does not overlap d.c. The power spectrum of the slicer input noise is not relevant to the following error probability calculation, which is dependent on only the statistics of a single sample of the noise. (Of course, noise correlation will cause errors to be dependent, but we do not quantify that phenomenon here.)

In this section we will assume that the slicer is designed according to the following principles:

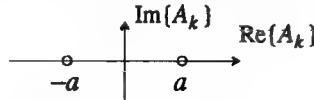
- It is *memoryless*, that is, it makes a symbol-by-symbol decision that considers only a single sample  $Q_k$  at a time.
- It applies the *minimum-distance criterion* of Section 7.2.1. That is, it chooses the data symbol  $a_j$  from the alphabet  $\{a_m, 1 \leq m \leq M\}$  that minimizes  $|Q_k - a_j|^2$ .

A *symbol error* occurs when the slicer chooses a data symbol  $a_j \neq a_m$ , where  $a_m$  was the actual transmitted symbol. Other definitions of error, such as bit error or block error, will be considered later.

The results of Section 8.2 apply directly to calculating the probability of symbol error. We can consider the complex data symbols and additive complex noise in two-dimensional Euclidean space. In the simplest case of a symbol alphabet of size two, let  $d$  be the distance between the symbols,  $d = |a_1 - a_2|$ . If either of the symbols is transmitted, the probability of the other symbol being chosen by the slicer is  $Q(d/2\sigma)$ .

#### Example 8-4.

Consider the binary signal constellation below, corresponding to a binary baseband PAM system:



This is called a *binary antipodal* signal constellation, and can be used for passband as well as baseband signaling. The distance between the symbols is  $d = 2a$ , and the probability of a symbol error is the same if either  $+a$  or  $-a$  is transmitted, so the probability of error is the same,

$$\Pr[\text{symbol error}] = Q(a/\sigma). \quad (8.55)$$

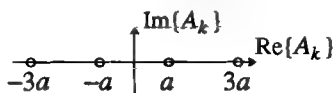
□

Fortunately, even in cases where  $M > 2$ , the probability of symbol error can often be calculated exactly, without resorting to the union bound, as will now be illustrated by

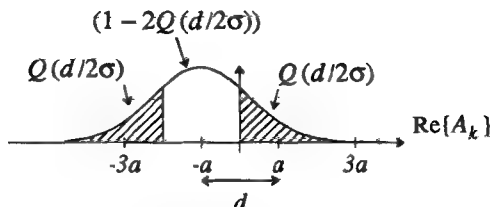
a series of examples.

### Example 8-5.

Consider the multilevel one-dimensional constellation shown in the following figure:



If the transmitted symbol at time  $k$  is  $A_k = -a$ , then the p.d.f. of  $Q_k$  is shown in the following figure:



The probability that the received sample  $Q_k$  is closer to a symbol other than  $-a$  is equal to the area of the shaded regions. The shaded regions each have area  $Q(d/2σ)$  so

$$\Pr[\text{symbol error at time } k \mid A_k = -a] = 2Q(d/2σ). \quad (8.56)$$

On the other hand,

$$\Pr[\text{symbol error at time } k \mid A_k = \pm 3a] = Q(d/2σ), \quad (8.57)$$

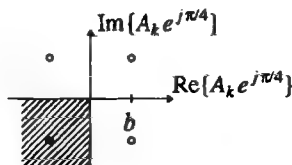
so if the symbols are equally likely at all times and independent then

$$\Pr[\text{symbol error}] = 1.5Q(d/2σ). \quad (8.58)$$

The coefficient 1.5 is the average number of nearest neighbors.  $\square$

### Example 8-6.

For the 4-PSK constellation of Figure 7-5a, the symbol  $-b$  can be mistaken for any of the other three if the noise is sufficiently large. The probability that it is mistaken for  $+jb$ , for example, is not precisely equal to the probability that it is closer to  $+jb$  than to  $-b$ , because it might be closest to  $+b$ . Denote the transmitted symbol by  $A_k$  and the received signal by  $Q_k = A_k + Z_k$ . For mathematical convenience, we rotate the coordinates through 45 degrees and rescale, so that the points are  $\{\pm b \pm jb\}$  as shown in Figure 8-1. For a correct decision,



**Figure 8-1.** A rotated version of the 4-PSK constellation in Figure 7-5. The shaded region is the decision region corresponding to one of the symbols.



the rotated  $Q_k'$  must lie in the shaded region in Figure 8-1. The statistics of the rotated noise are the same as the statistics of the non-rotated noise, since the noise is circularly symmetric. Then

$$M_k = Z_k e^{j\pi/4} \quad (8.59)$$

is a complex-valued Gaussian noise with independent real and imaginary parts, each with variance  $\sigma^2$ . The probability of  $Q_k'$  lying in the shaded region is thus

$$\begin{aligned} \Pr[\text{correct decision} | A_k] &= \Pr[\text{Re}\{M_k\} < b, \text{Im}\{M_k\} < b] \\ &= \Pr[\text{Re}\{M_k\} < b] \Pr[\text{Im}\{M_k\} < b] \\ &= (1 - Q(d/2\sigma))^2 \\ &= 1 - 2Q(d/2\sigma) + Q^2(d/2\sigma), \end{aligned} \quad (8.60)$$

where  $d = 2b$ , the minimum distance between symbols. The probability of a symbol error is simply

$$\begin{aligned} \Pr[\text{symbol error} | A_k] &= 1 - \Pr[\text{correct decision} | A_k] \\ &= 2Q(d/2\sigma) - Q^2(d/2\sigma). \end{aligned} \quad (8.61)$$

The probability of error is the same for whatever symbol is transmitted, so the overall probability of symbol error is the same as the conditional probability of error  $\Pr[\text{symbol error} | A_k]$ . It is common to make the simple approximation that the noise is small enough (the SNR is large enough) that  $Q(d/2\sigma)$  is small. In this case

$$\Pr[\text{symbol error}] \approx 2Q(d/2\sigma). \quad (8.62)$$

This approximation is reasonable, because communication is likely to be useful only if the probability of error is small, or  $Q^2(d/2\sigma)$  is small compared to  $Q(d/2\sigma)$  (see Problem 8-3). Here the coefficient "2" reflects the fact that every symbol has two nearest neighbors.  $\square$

### Example 8-7.

The probability of error for the 16-QAM constellation in Figure 7-5 can be found by a similar method. Consider the four inside points. Their decision regions are squares with sides equal to  $d$ . In this case, the probability of a correct decision is

$$\Pr[\text{correct decision} | A_k \text{ on the inside}] = [1 - 2Q(d/2\sigma)]^2 \quad (8.63)$$

so the probability of error is

$$\Pr[\text{error} | A_k \text{ on the inside}] = 4Q(d/2\sigma) - 4Q^2(d/2\sigma) \approx 4Q(d/2\sigma), \quad (8.64)$$

consistent with the fact that every interior point has four nearest neighbors. The probability of error of the corner symbols is similar to the probability of error for the 4-PSK signal,

$$\Pr[\text{error} | A_k \text{ in the corner}] \approx 2Q(d/2\sigma), \quad (8.65)$$

and

$$\Pr[\text{error} | A_k \text{ not inside or corner}] \approx 3Q(d/2\sigma). \quad (8.66)$$

Assuming the symbols are equally likely, the total probability of error is

$$\begin{aligned}\Pr[\text{error}] &\approx \frac{4}{16} \times 4Q(d/2\sigma) + \frac{8}{16} \times 3Q(d/2\sigma) + \frac{4}{16} 2Q(d/2\sigma) \\ &= 3Q(d/2\sigma).\end{aligned}\quad (8.67)$$

The exact probability of error (without the high SNR assumption) can be easily computed (see Problem 8-5).  $\square$

For rectangular two-dimensional constellations, as the size of the alphabet gets large, a greater percentage of the points will be in the interior with four nearest neighbors, and the probability of error will tend towards

$$\Pr[\text{symbol error}] \rightarrow 4Q(d/2\sigma) - 4Q^2(d/2\sigma), \quad (8.68)$$

assuming all symbols are equally likely.

### Union Bound

In the above analysis we found an expression for the exact probability of error and then got a simpler approximation by neglecting insignificant terms. This technique works well for many examples, but not all. A more general (and approximate) analysis uses the union bound described earlier. To apply the union bound, for a particular transmitted symbol, the error probability is upper-bounded by the sum of the probabilities that each of the other symbols considered in isolation is chosen.

#### Example 8-8.

For the 4-PSK constellation of Example 8-6, by symmetry the error probability will be the same regardless of which symbol is transmitted. For each symbol, there are two symbols at distance  $d$  and one at distance  $\sqrt{2}d$ . Each of the symbols at distance  $d$  would be chosen with probability  $Q(d/2\sigma)$ , if it were the only other symbol, and similarly the symbol at distance  $\sqrt{2}d$  would be chosen with probability  $Q(d/\sqrt{2}\sigma)$ . Thus, the union bound is

$$\Pr[\text{symbol error} | A_k] \leq 2 \cdot Q(d/2\sigma) + Q(d/\sqrt{2}\sigma). \quad (8.69)$$

The first term will tend to dominate at moderate to high SNR (see Problem 8-4), so this bound is a good approximation (compare to (8.61)).  $\square$

The union bound will be particularly useful in Chapters 13 and 14 in analyzing the performance of coded systems.

### 8.3.3. Other Error Measures

In the previous section, we have determined the probability of symbol error, defined as the probability that an incorrect data symbol is substituted for the correct one. In actuality, the data symbols represent a group of bits. At the receiver, a symbol error is translated into one or more bit errors. In this section, we are interested in evaluating the probability of a bit error. We are also interested in the probability of a block error, which is defined as one or more errors in a block of bits.

## Probability of Bit Error

In some applications, the quality of a digital communication system is measured by the probability of a *bit* error rather than a *symbol* error. Bit errors are caused by symbol errors, but the exact relationship between their probabilities depends on the coder (that maps input bits into symbols.) If the SNR is high enough, as it is for most useful communication systems, then a symbol is far more likely to be mistaken for one of its neighbors in the constellation than for more distant symbols (see for example Problem 8-4). Coders often implement a *Gray code* as in Figure 8-2, in which nearest neighbors correspond to bit groups that differ by only one bit. Thus, the most probable symbol errors cause only a single bit error. If the SNR is large, then

$$\Pr[\text{bit error}] \approx \frac{1}{M} \Pr[\text{symbol error}] \quad (8.70)$$

where  $M$  is the number of bits per symbol.

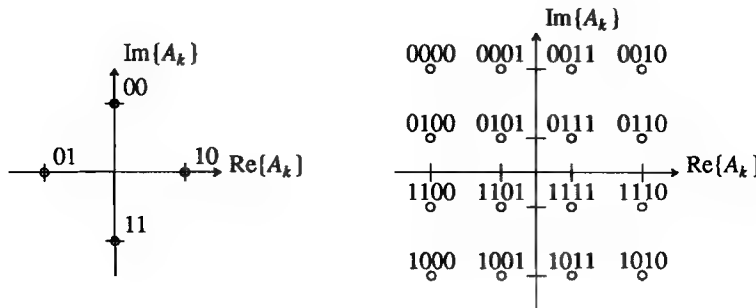
In many cases,  $\Pr[\text{bit error}]$  can be calculated directly and exactly, without resorting to the approximation of (8.70).

## Probability of Block Error

Often, a modem is embedded in a communication system that transmits blocks of bits, performs a rudimentary error check, and retransmits the block if an error is detected. In this case, the probability of bit error is of less interest than the *probability of block error*. Let  $B$  denote the number of bits in a block, so  $B/M$  is the number of symbols in a block. Let  $\Pr[\text{symbol error}]$  denote the probability of symbol error. If symbol errors are independent, then a block is entirely correct with probability

$$(1 - \Pr[\text{symbol error}])^{B/M} \quad (8.71)$$

so



**Figure 8-2.** The two constellations of Figure 7-5 are shown with a set of bits associated with each symbol. Notice that only one bit differs between any two adjacent symbols. This minimizes the number of bit errors per symbol error. This type of code is called a Gray code.

$$\Pr[\text{block error}] = 1 - (1 - \Pr[\text{symbol error}])^{B/M} . \quad (8.72)$$

If  $\Pr[\text{symbol error}]$  is small, then its higher powers can be neglected and

$$\Pr[\text{block error}] \approx \frac{B}{M} \Pr[\text{symbol error}] . \quad (8.73)$$

#### Example 8-9.

Suppose  $B = 1000$ ,  $M = 4$ , and  $\Pr[\text{symbol error}] = 10^{-6}$ . Then  $\Pr[\text{block error}] \approx 2.5 \times 10^{-4}$ . These parameters are typical.  $\square$

## 8.4. PERFORMANCE of MINIMUM-DISTANCE RECEIVERS

The last section analyzed the error probability for baseband and passband PAM using the receiver structure derived on intuitive grounds in Chapter 6. In Chapter 7, an alternative receiver design methodology, minimum-distance receiver design, was demonstrated. This section will derive the error probability of minimum-distance receivers, using the fundamental results of Section 8.2.

### 8.4.1. Baseband-Equivalent Noise

We are interested in the effect of noise introduced at passband on receivers that use minimum-distance receiver design criteria. A formulation of the receiver design expressed in terms of passband signals is thus needed.

#### Receiver Structure

Assume that  $Y(t)$  is a real-valued passband received signal, given by

$$Y(t) = \sqrt{2} \operatorname{Re}\{s_m(t)e^{j\omega_c t}\} + N(t) , \quad (8.74)$$

where  $N(t)$  is real-valued white Gaussian noise with power spectral density  $N_0$  and  $s_m(t)$  is the received complex baseband signal drawn from a set of  $L$  known signals  $\{s_l(t), 1 \leq l \leq L\}$ . The known signals can be expressed in terms of set of  $N$  complex-valued orthonormal basis functions

$$s_l(t) = \sum_{n=1}^N S_{l,n} \phi_n(t) . \quad (8.75)$$

Since the set of known signals is assumed to be passband in nature, assume that each  $s_l(t)$  and each  $\phi_n(t)$  is bandlimited to less than  $\omega_c$  radians/sec.

As shown in Chapter 7, the minimum-distance receiver calculates the set of decision variables

$$C_n = \int_{-\infty}^{\infty} Y(t) \sqrt{2} e^{-j\omega_c t} \phi_n^*(t) dt , \quad 1 \leq n \leq N , \quad (8.76)$$

and then chooses the signal with index  $l$  that satisfies the Euclidean-distance criterion

$$\min_l \sum_{n=1}^N |C_n - S_{l,n}|^2. \quad (8.77)$$

Analyzing the performance of this receiver requires a statistical characterization of the decision variables  $C_n$ ,  $1 \leq n \leq N$ .

### Statistics of the Decision Variables

Substituting (8.74) into (8.76), the decision variables are given in terms of the signal and noise as

$$C_n = S_{m,n} + Z_n \quad (8.78)$$

where

$$S_{m,n} = 2 \cdot \int_{-\infty}^{\infty} \operatorname{Re}\{s_m(t)e^{j\omega_c t}\} e^{-j\omega_c t} \phi_n^*(t) dt \quad (8.79)$$

$$Z_n = \int_{-\infty}^{\infty} N(t) \sqrt{2} e^{j\omega_c t} \phi_n^*(t) dt. \quad (8.80)$$

#### Exercise 8-1.

Verify that the integral on the right side of (8.79) is in fact equal to  $S_{m,n}$  as defined by (8.75) when  $l = m$ .  $\square$

The equivalent noise  $Z_n$  is a complex-valued zero-mean Gaussian random process. Using precisely the same techniques as in Section 8.3, the orthogonality of the basis functions  $\phi_n(t)$  implies that the noise samples are uncorrelated and have the same variance,

$$E[Z_i Z_j^*] = 2\sigma^2 \delta_{i,j}, \quad 1 \leq i, j \leq N, \quad (8.81)$$

where  $\sigma^2 = N_0$ , due to the unit energy of  $\phi_n(t)$ . Further, the fact that the  $\phi_n(t)$  are bandlimited to  $\omega_c$  implies that the  $Z_n$  are circularly symmetric,

$$E[Z_i Z_j] = 0, \quad 1 \leq i, j \leq N. \quad (8.82)$$

Together, these properties establish that the  $Z_n$  are mutually independent and identically distributed, with identically distributed and independent real and imaginary parts.

In fact, the decision variables of (8.78) represent a vector-valued received signal represented by  $\mathbf{C}' = [C_1, C_2, \dots, C_N]$  that is mathematically identical to (8.20),

$$\mathbf{C} = \mathbf{S}_m + \mathbf{Z}. \quad (8.83)$$

including the same statistics for the noise vector  $\mathbf{Z}$ . Furthermore, the minimum-distance criterion of (8.77) is mathematically equivalent to (8.21),

$$\min_l \|\mathbf{C} - \mathbf{S}_l\|^2. \quad (8.84)$$

Therefore, the results of Section 8.2 apply directly to the minimum-distance receiver design. All the work is already done!

To summarize the earlier results, from the upper and lower bounds we conclude that for small  $\sigma$  the probability of the minimum-distance receiver choosing a signal different from that transmitted is approximately

$$P_e \approx K \cdot Q\left(\frac{d_{\min}}{2\sigma}\right), \quad (8.85)$$

where  $K$  is a constant and  $d_{\min}$  is the minimum-distance in the signal set.

The parameter  $d_{\min}$  was already determined in Chapter 7 for several modulation techniques. In that chapter, it was argued that  $d_{\min}$  is a measure of the noise immunity. This is confirmed by (8.85), which shows further that it is the size of  $d_{\min}$  in relation to the noise standard deviation  $\sigma$  that matters most. The coefficient  $C$  (which is the average number of nearest neighbors) is secondary.

### 8.4.2. Probability of Error for Minimum-Distance Design

Determining the probability of error for different signaling schemes is now a simple matter of substituting for  $d_{\min}$  in (8.85). In addition, as in the case of the slicer, we will show that the exact probability of error can often be determined without resorting to the approximation of (8.85).

It is difficult to use probability of error to compare different modulation schemes, because we typically want to keep the transmit signal powers the same, or the spectral efficiencies the same, or some similar constraint. Thus, we defer a comparison of the performance of different modulation schemes to Section 8.7, where spectral and power efficiency are taken into account.

#### Isolated Pulse PAM with Matched Filter

For the detection of PAM with an isolated pulse  $h(t)$ , the minimum-distance criterion resulted in a matched filter followed by sampler and slicer, as in Figure 7-6. The resulting input to the slicer was the data symbol multiplied by  $\sigma_h$ , where  $\sigma_h^2$  is the energy in the received pulse  $h(t)$ . The minimum distance is thus the same as the minimum distance of the data symbol alphabet, already considered in Section 8.3, multiplied by  $\sigma_h$ . Calling the symbol alphabet minimum-distance  $a_{\min}$ , the error probability is approximately

$$P_e \approx K \cdot Q\left(\frac{\sigma_h a_{\min}}{2\sigma}\right). \quad (8.86)$$

This is the same as the slicer design error probability considered earlier, but with a different scaling of signal level.

#### Orthogonal Multipulse

For orthogonal multipulse, each signal is of the form

$$\mathbf{S}_m = [0, 0, \dots, \sigma_h, 0, \dots, 0] \quad (8.87)$$

where the non-zero term is in the  $m$ -th position. Thus, every signal is the same distance from every other signal, namely  $d = \sqrt{2}\sigma_h$ , so the minimum distance is  $d_{\min} = \sqrt{2}\sigma_h$ . For  $N$  orthogonal signals, there are  $N-1$  other signals at the minimum distance, and the error probability is independent of which signal is transmitted. The error probability is approximated by

$$P_e \approx (N-1) \cdot Q\left(\frac{\sqrt{2}\sigma_h}{\sigma}\right). \quad (8.88)$$

The argument of  $Q(\cdot)$  is not a function of  $N$ , but the error probability does increase slowly with  $N$  because of the factor  $N-1$  multiplying  $Q(\cdot)$ . The basic tradeoff is that, at the expense of more bandwidth and with only a minor penalty in error probability, the number of bits per symbol can be increased by increasing  $N$ . It is surprising that the required increase in bandwidth associated with increasing  $N$  does not result in a greater penalty in error probability, since increasing bandwidth is usually associated with allowing more noise into the receiver. This issue will be considered further in Section 8.5 (in the context of spread spectrum), where it will be emphasized that increasing bandwidth does not increase the noise when a matched filter is used.

The exact error probability can be calculated with relative ease because of the simplicity of the signal geometry. In particular, we showed in Section 7.2.3 that the minimum-distance receiver simplifies, for this geometry, to the criterion

$$\max_l \operatorname{Re}\{C_l\}, \quad (8.89)$$

or in words, the receiver chooses the largest cross-correlation between the received signal and the orthogonal pulses. The error probability does not depend on which signal is transmitted, so assume that it is  $S_1$ . In that case, all the  $\operatorname{Re}\{C_l\}$  are independent Gaussian random variables with variance  $\sigma^2$ , and all are zero-mean except for  $\operatorname{Re}\{C_1\}$ , which has mean  $\sigma_h$ . A correct decision is made if  $\operatorname{Re}\{C_1\}$  is larger than  $\operatorname{Re}\{C_l\}$ ,  $2 \leq l \leq N$ . Thus,

$$\Pr[\text{correct decision} \mid S_1 \text{ transmitted}, \operatorname{Re}\{C_1\} = \alpha] = (1 - Q(\frac{\alpha}{\sigma}))^{N-1}, \quad (8.90)$$

and

$$\begin{aligned} \Pr[\text{error} \mid S_1 \text{ transmitted}] &= 1 - \Pr[\text{correct decision} \mid S_1 \text{ transmitted}] \\ &= 1 - \int_{-\infty}^{\infty} f_{\operatorname{Re}\{C_1\}}(\alpha) (1 - Q(\frac{\alpha}{\sigma}))^{N-1} d\alpha \end{aligned} \quad (8.91)$$

where  $f_{\operatorname{Re}\{C_1\}}(\alpha)$  is the p.d.f. of a Gaussian random variable with mean  $\sigma_h$  and variance  $\sigma^2$ ,

$$f_{\operatorname{Re}\{C_1\}}(\alpha) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\alpha - \sigma_h)^2/2\sigma^2}. \quad (8.92)$$

Since this error probability will be the same regardless of which signal is transmitted,

$$\Pr[\text{error} \mid S_1 \text{ transmitted}] = \Pr[\text{error}]. \quad (8.93)$$

The integral in (8.91) does not have a closed form solution, but is tabulated in [3] and plotted in Figure 8-3. Notice that for large SNR, the error probability is only weakly dependent on  $N$ , the number of orthogonal pulses, as predicted by the error probability approximation. Going from  $N = 2$  to  $N = 95$  results in less than a 2 dB penalty.

### Combined PAM and Orthogonal Multipulse

In combined PAM and multipulse, if each of the orthogonal pulses is independently modulated by data symbols drawn from the same alphabet, and the minimum-distance for that alphabet is  $a_{\min}$ , then  $d_{\min} = \sigma_h a_{\min}$ . The probability of error is thus equivalent to (8.86),

$$P_e \approx K \cdot Q\left(\frac{\sigma_h a_{\min}}{2\sigma}\right). \quad (8.94)$$

Assuming they both use the same data symbol alphabet, PAM and combined PAM and orthogonal multipulse have approximately the same error probability at high SNR. We saw in Chapter 6 that their spectral efficiencies are essentially the same as well.

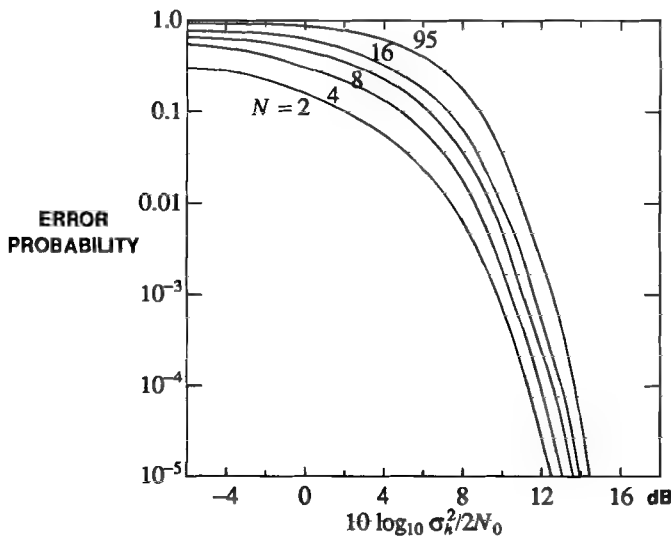


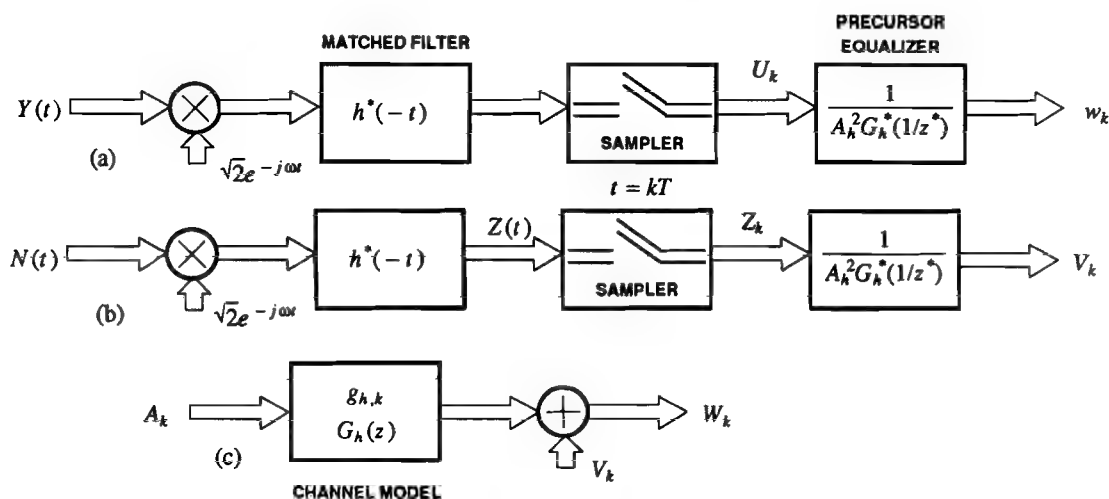
Figure 8-3. Error probability for  $N$  orthogonal pulses with energy  $\sigma_h^2$  [4].



## 8.5. PAM with ISI

In Section 8.3 the noise performance of PAM for the intuitive design of Chapter 6 was considered. In Section 8.4 this was extended to the minimum-distance receiver design, but only for the isolated pulse case. In this section, we analyze the noise performance of the minimum-distance receiver design of Figure 7-11. Rather than using the receive filter to eliminate ISI, as in Chapter 6, this minimum-distance receiver equalizes only the anticausal portion of the ISI, and then uses a Euclidean distance minimization to determine the sequence of data symbols. In this section, we will consider this minimum-distance receiver structure from the perspective of noise. One striking conclusion will be that the noise at the output of the sampled matched filter is nonwhite, but the noise at the output of the precursor equalizer is white. This is no accident, since it will be shown in Chapter 9 that the precursor equalizer is actually an optimal linear predictor, whitening the noise as well as eliminating the anticausal portion of the ISI. The signal path of Figure 7-11 is repeated in Figure 8-4a, with the addition of the demodulator for completeness. The matched filter, symbol-rate sampler, and precursor equalizer, are together called a *whitened matched filter (WMF)*. This is because, as we will show shortly, the noise at the WMF output is white.

This gives an additional interpretation, of the discrete-time precursor equalizer. One of its functions is to turn the two-sided isolated pulse response  $p_h(k)$  at the output of the sampled matched filter into a causal response  $g_{h,k}$ . A second function of the precursor equalizer is to whiten the noise, which is not white at the output of the



**Figure 8-4.** The whitened matched filter for detection of PAM signals with ISI. a) The front-end filtering and sampling, b) the response to white Gaussian noise on the channel, and c) the equivalent discrete-time model.

sampled matched filter. For this reason, the precursor equalizer is also called a *discrete-time whitening filter*.

### Response of WMF to Data Symbols

Consider the response of the WMF to a single isolated pulse  $h(t)$ . The pulse at the output of the sampler is  $\rho_h(k)$ , the sampled autocorrelation function of  $h(t)$ . Since its Z-transform, the folded spectrum  $S_h(z)$ , has the factorization  $S_h(z) = A_h^2 G_h(z) G_h^*(1/z^*)$ , the output of the precursor equalizer has Z-transform  $G_h(z)$ . Thus, the signal path has an equivalent discrete-time transfer function  $G_h(z)$  to data symbols, as shown in the discrete-time model of Figure 8-4c. This equivalent response  $G_h(z)$  is causal, and this is the origin of the term "precursor equalizer". Among other things, the precursor equalizer eliminates or "equalizes" the noncausal portion of the ISI, called the precursor. It turns a two-sided impulse response  $\rho_h(k)$  at the sampled matched filter output into a causal response  $g_{h,k}$  at the WMF output.

### Noise at Output of WMF

The noise statistics at the WMF output can be determined using Figure 8-4b, with the final result that the noise is white, as shown in Figure 8-4c. For a zero-mean white Gaussian input noise  $N(t)$  with power spectrum  $N_0$  on the passband channel, the statistics of the noise at the matched filter output were characterized in Section 8.3. In particular, the complex-valued noise  $Z(t)$  at that point is a stationary circularly symmetric zero-mean Gaussian process with power spectrum

$$S_Z(j\omega) = 2N_0 |H(j\omega)|^2, \quad (8.95)$$

since the receive filter is  $f(t) = h^*(-t)$ . Since the process is circularly symmetric, it is fully characterized by this power spectrum. A sampled version of this noise  $Z_k$  is also circularly symmetric, and has power spectrum

$$S_Z(e^{j\omega T}) = 2N_0 S_h(e^{j\omega T}), \quad (8.96)$$

where again the folded spectrum arises. Note that this noise at the sampled matched filter output is nonwhite in general. It is white if and only if the folded spectrum is flat, which is precisely the Nyquist criterion.

Finally, the noise component at the output of the WMF,  $V_k$ , is Gaussian, circularly symmetric (since it is a filtered circularly symmetric process), and has power spectrum

$$S_V(z) = 2N_0 S_h(z) \cdot \frac{1}{A_h^2 G_h^*(1/z^*)} \cdot \frac{1}{A_h^2 G_h(z)} = \frac{2N_0}{A_h^2}. \quad (8.97)$$

As promised, the output noise is white. Furthermore, the samples of this noise are statistically independent, since it is white and circularly symmetric. In addition, the real- and imaginary parts are statistically independent, and have the same variance

$$\sigma^2 = \frac{N_0}{A_h^2}. \quad (8.98)$$

There is also expression (2.57) for  $A_h^2$ , since it arises in the minimum-phase spectral

factorization of  $S_h(z)$ . This yields an expression for the noise variance,

$$\sigma^2 = N_0 \cdot \exp \left\{ -\frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \ln [S_h(e^{j\omega T})] d\omega \right\}. \quad (8.99)$$

For rational folded spectra, it is not necessary to evaluate this geometric mean integral, since the constant  $A_h^2$  can be read directly from the rational spectrum. This is illustrated by two examples.

#### Example 8-10.

For the example received pulse  $h(t)$  in Example 7-10, the folded spectrum is first-order all-pole and rational,

$$S_h(z) = \sigma_h^2(1 - \alpha^2) \cdot \frac{1}{1 - \alpha z^{-1}} \cdot \frac{1}{1 - \alpha z}. \quad (8.100)$$

This is written in the form of a minimum phase spectral factorization, since the first term  $G_h(z) = 1/(1 - \alpha z^{-1})$  is minimum-phase and monic ( $G_h(\infty) = 1$ ). Because each term is monic, we deduce that

$$A_h^2 = \sigma_h^2(1 - \alpha^2). \quad (8.101)$$

without the need to evaluate the geometric-mean integral.  $\square$

#### Example 8-11.

For the received pulse  $h(t)$  in Example 7-11,

$$S_h(z) = \sigma_0^2(1 + \alpha z^{-1})(1 + \alpha z). \quad (8.102)$$

Again, this is written in the form of a monic minimum-phase spectral factorization, and we identify  $A_h^2 = \sigma_0^2$ .  $\square$

### Minimum Distance Receiver

Combining the results thus far, the output of the WMF is

$$W_k = \sum_{m=1}^K a_m g_{h,k-m} + V_k \quad (8.103)$$

where  $V_k$  is white Gaussian noise with variance  $2\sigma^2 = 2N_0/A_h^2$ . This has the same form as the problem posed in (8.20), where the known signal has components

$$s_k = \sum_{m=1}^K a_m g_{h,k-m}, \quad 1 \leq k < \infty. \quad (8.104)$$

In this case, the signal has in general an infinite number of components, rather than the finite number of components assumed in (8.20), unless  $G_h(z)$  happens to be FIR, in which case (8.104) will be finite in extent. The minimum-distance receiver design then performs the minimization

$$\min_{\{a_k, 1 \leq k \leq K\}} \sum_{k=1}^{\infty} |W_k - \sum_{m=1}^K a_m g_{h,k-m}|^2. \quad (8.105)$$

This is equivalent to the criterion of (7.61).

The minimum-distance receiver criterion of (8.105) has been derived *assuming* that a WMF was the receiver front end. It was shown in Section 7.3.2 (see (7.61)) that this discrete-time criterion is equivalent (in the sense that it will choose the same sequence  $\{a_k, 1 \leq k \leq K\}$ ) to a continuous-time minimum-distance criterion. The matched filter turns the continuous-time received signal into a symbol-rate discrete-time signal, but in the process colors the noise and also introduces ISI. The precursor equalizer serves two purposes simultaneously: first, it equalizes the anticausal portion of the ISI, and second, it whitens the noise.

We will show in Chapter 9 that the minimum-distance criterion (in either continuous or discrete time) is optimal for Gaussian noise on the channel (according to a particular definition of optimality), and we will in the process extend the WMF to nonwhite Gaussian noise on the channel.

### Probability of Error

From Section 8.2, the probability of error is approximately

$$P_e \approx K \cdot Q\left(\frac{d_{\min}}{2\sigma}\right). \quad (8.106)$$

Since our interpretation of the "set of known signals" in this case is the set of all PAM signals corresponding to a *sequence* of  $K$  data symbols, the interpretation of (8.106) is the *probability of sequence error*. By sequence error, we mean *one or more* errors in the detection of the entire sequence of data symbols. The minimum distance  $d_{\min}$  is the minimum Euclidean distance between two distinct sequences of data symbols,

$$d_{\min} = \min_{\{\epsilon_k, 1 \leq k \leq K\}} \sum_{m=1}^{\infty} \left| \sum_{k=1}^K \epsilon_k g_{h,m-k} \right|^2, \quad (8.107)$$

where  $\{\epsilon_k, 1 \leq k \leq K\}$  is a non-zero sequence of "error symbols",  $\epsilon_k = a_k^{(1)} - a_k^{(2)}$ . The minimization is over all sequences of  $a_k^{(1)}$  and  $a_k^{(2)}$  that are not equal; that is, that differ for at least one  $k$ . Within a constant (which is due to the different normalization), this minimum distance is equal to the minimum distance calculated directly in continuous-time in Section 7.3. A practical algorithm for calculating the minimum-distance of (8.107) will be explained in Chapter 9.

## 8.6. SPREAD SPECTRUM

Spread spectrum, a term applied to passband PAM when the bandwidth is chosen to be very much larger than the minimum dictated by the Nyquist criterion, was briefly introduced in Section 6.7. Spread-spectrum has a long history, mostly in secure military communications, as discussed by Scholtz [5]. More recently, a number of commercial applications have arisen, for example in digital cellular systems. A useful definition of spread-spectrum is [6]:

Spread-spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the

band spread is accomplished by means of a code that is independent of the data, and a synchronized reception with the code at the receiver is used for de-spreading and subsequent data recovery.

The *spreading code* used in this definition will be defined shortly.

### Bandwidth and Probability of Error

Assume the channel noise is white and Gaussian. If we are to increase the bandwidth of the received pulse  $h(t)$ , the question is whether this bandwidth expansion adversely affects the SNR at the slicer. Intuitively it might, because we must expand the bandwidth of the receive filter and let in more noise. This logic is valid with a simple lowpass filter in the receiver. However, with a matched filter, there is *no* relationship between bandwidth and slicer SNR.

To show this, consider an isolated pulse input to the receiver, which consists of a matched filter and sampler. For received signal  $A_k h(t)$ , the single signal sample at the matched filter output is  $A_k \sigma_h^2$ , which has minimum-distance  $d_{\min} = \sigma_h^2 a_{\min}$ . For white noise with spectral density  $N_0$ , the matched filter output noise has variance  $\sigma^2 = \sigma_h^2 N_0$ . Thus the error probability is approximately

$$P_e \approx K \cdot Q\left(\frac{\sigma_h^2 a_{\min}}{2\sigma_h \sqrt{N_0}}\right) = K \cdot Q\left(\frac{\sigma_h a_{\min}}{2\sqrt{N_0}}\right). \quad (8.108)$$

$P_e$  depends on the *energy* of the received pulse, but not its bandwidth.

An intuitive explanation of this bandwidth independence is as follows. By the Landau-Pollak theorem (Section 7.4), the space of received pulses bandlimited to  $B$  Hz and approximately time-limited to  $T$  sec (the symbol interval) has approximate dimension  $2BT$ . By definition, spread spectrum corresponds to  $2BT \gg 1$ , where this approximation becomes accurate. Since  $h(t)$  only occupies one dimension, the matched filter captures only a fraction  $1/2BT$  of the total noise in bandwidth  $B$ . While the variance of this total noise is proportional to  $B$ , on net, the noise variance at the output of matched filter is not dependent on  $B$ .

It is common in practice to characterize the signal-to-noise ratio (SNR) at the receiver input, in preference to the energy per bit and noise power spectral density. A formula for  $P_e$  based SNR will also prove valuable in the next section. Assuming no ISI (translates  $h(t - kT)$  are mutually orthogonal), the received signal power,  $P_S$ , is equal to the energy per symbol ( $\sigma_A^2 \sigma_h^2$ ) times the symbol rate  $1/T$ ,

$$P_S = \sigma_A^2 \sigma_h^2 / T. \quad (8.109)$$

Furthermore, the total noise power within bandwidth  $B$  is  $2N_0B$ . The received SNR is defined as the ratio of signal power to noise power,

$$SNR = P_S / 2N_0B. \quad (8.110)$$

Substituting into (8.108),  $P_e$  expressed in terms of  $SNR$  is

$$P_e \approx K \cdot Q(\sqrt{2BT \cdot \eta_A \cdot SNR}). \quad (8.111)$$

The quantity

$$\eta_A = a_{\min}^2 / 4\sigma_A^2 \quad (8.112)$$

is a parameter of the signal constellation, and is independent of any scaling of that constellation. If we keep  $SNR$  constant, then  $P_e$  decreases as the dimensionality  $2BT$  increases. However, in order to keep  $SNR$  constant for a fixed  $N_0$ ,  $P_S$  has to be increased in proportion to  $B$ . If  $P_S$  is kept fixed, then  $P_e$  is independent of  $B$  as stated earlier.

The bandwidth independence of  $P_e$  for fixed signal power  $P_S$  presumes a matched filter in the receiver. We saw in Section 8.4.3 that the matched filter maximizes the  $SNR$  at the slicer, and thus a different receive filter will inevitably result in a lower  $SNR$  and higher  $P_e$ . With spread spectrum, the use of a different receive filter can be disastrous. As  $2BT$  increases, there are an increasing number of waveforms that are bandlimited to  $B$ , approximately time limited to  $T$ , and orthogonal to the actual pulse  $h(t)$ . If we happen to use a filter matched to one of these, the signal component at the receive filter output will be zero! This same observation also explains why spread spectrum has been used for the concealment of communications in military applications.

### Generating Broadband Pulses

Spread spectrum requires ways to generate broadband pulses with controlled spectral properties. A whole family of pulse shapes  $h(t)$ , each with the same amplitude spectrum and different phase spectra, is conveniently generated using a *chip waveform* and *spreading sequence*. In this approach, the symbol interval  $T$  is divided into  $N$  sub-intervals, each of duration  $T_c = T/N$ . Within each sub-interval, a pulse-amplitude modulated time translate of a pulse  $h_c(t)$  is transmitted. The translate of  $h_c(t)$  is called a *chip*. The pulse  $h(t)$  is formed from a PAM modulation of the chips by some deterministic sequence  $\{x_m, 0 \leq m \leq N-1\}$ , called the *spreading sequence*,

$$h(t) = \sum_{m=0}^{N-1} x_m h_c(t - mT_c), \quad H(j\omega) = H_c(j\omega) \sum_{m=0}^{N-1} x_m e^{-j\omega mT_c}. \quad (8.113)$$

The bandwidth of the resulting pulse will equal the bandwidth of  $h_c(t)$ . Typically, we choose  $h_c(t)$  to satisfy the Nyquist criterion at pulse rate  $1/T_c$ , which requires a minimum bandwidth of  $B = 1/2T_c = N/2T$  Hz; this causes a bandwidth expansion by a factor of  $N = T/T_c$ . The spectrum can be controlled to some degree by the spreading sequence, with precisely  $N$  degrees of freedom.

The chip waveform and spreading sequence can also be used to generate orthogonal pulses (see Problem 8-11). For example, such orthogonal pulses are required for CDMA systems (Section 6.9).

### ISI and Spread Spectrum

The preceding error probability calculation presumed no ISI. In fact, spread spectrum affords a degree of immunity to ISI. To understand this, we need to consider the pulse shape at the output of a receiver matched filter, and then the effect of channel dispersion.

Keeping the symbol interval  $T$  fixed, and increasing the bandwidth  $B$ , the ISI in the transmit pulse can be reduced. For a large  $2BT$ , a pulse bandlimited to  $B$  can be largely confined to an interval  $T$ , in the sense that a diminishing fraction of the pulse energy falls outside that interval as  $2BT$  increases. (In fact, approximately  $2BT$  orthogonal pulses can satisfy this condition simultaneously.) When  $2BT$  is near unity, even a single pulse cannot come close to being time-limited to  $T$ . Thus, the *transmit* pulse can come much closer to being time-limited in a spread-spectrum system.

However, the *receive* pulse is affected by the channel; furthermore, we are interested in the ISI at the matched filter output rather than the channel output. (Recall that the matched filter is crucial to the operation of a spread spectrum system because of its power to suppress in-band noise.) Assume for the moment that the channel is ideal. The isolated-pulse output of the matched filter is then the pulse autocorrelation function,  $\rho_h(t)$ . The Fourier transform of that isolated pulse is  $|H(j\omega)|^2$ , which by definition has a wide bandwidth  $B$ .

### Example 8-12.

Suppose that  $|H(j\omega)|^2$  is constant over the bandwidth  $B$  and zero elsewhere. If we normalize the energy of  $h(t)$  to unity, then  $|H(j\omega)|^2 = 1/2B$ . The isolated pulse at the matched filter output is

$$\rho_h(t) = \text{sinc}(2\pi Bt). \quad (8.114)$$

As  $B$  increases, the energy of this pulse concentrates in a shorter time duration. Furthermore, if  $2BT$  is an integer,  $\rho_h(t)$  always obeys the Nyquist criterion,

$$\rho_h(kT) = \text{sinc}(k\pi \cdot 2BT) = \delta_k. \quad (8.115)$$

□

This simple example illustrates two important points:

- For an ideal channel, the isolated pulse at the output of the matched filter is dependent only on the magnitude spectrum of  $h(t)$ , and is not dependent on the phase spectrum. Even though we have specified the magnitude spectrum in Example 8-12, there remains flexibility in choosing the phase.
- The time duration of isolated pulse at the output of the matched filter can be much shorter than the symbol interval, even though  $h(t)$  completely fills the symbol interval. The greater  $B$ , the shorter this duration can be.

Pulses of the form of (8.113) can be designed to have a narrow autocorrelation function. The pulse autocorrelation (matched filter output isolated pulse) will conform to Example 8-12 if two sufficient (but not necessary) conditions are satisfied:

- The chip pulse  $h_c(t)$  is an ideal LPF with bandwidth  $B = 1/2T_c$ ,  $h_c(t) = \text{sinc}(\pi t/T_c)$ , and
- The sequence  $\{x_m\}$  is chosen to satisfy

$$\left| \sum_{m=0}^{N-1} x_m e^{-j\omega m T_c} \right|^2 = 1 \quad \text{for all } \omega. \quad (8.116)$$

**Example 8-13.**

A trivial case is  $x_k = \delta_{k-L}$  for some  $0 \leq L \leq N-1$ . Regardless of  $L$ , (8.116) is satisfied. The choice of  $L$  affects the phase spectrum of  $h(t)$ , but not the magnitude spectrum. The problem with this choice is that the peak signal is very large in relation to  $\sigma_h^2$ , creating practical difficulties on most channels, and especially on radio channels.  $\square$

**Example 8-14.**

We can increase  $\sigma_h^2$  for a given peak signal by choosing  $|x_m|^2 = 1/N$ ,  $0 \leq m \leq N-1$ . For such choices, (8.116) cannot be exactly satisfied. However, a good approximate approach is to force (8.116) to be satisfied at uniformly spaced frequencies, with spacing  $2\pi/NT_c$ ,

$$\left| \sum_{m=1}^N x_k e^{-j2\pi mn/N} \right|^2 = 1/N, \quad 0 \leq n \leq N-1. \quad (8.117)$$

This is the condition that the DFT of  $\{x_m\}$  have constant magnitude.  $\square$

It is possible to come very close to satisfying the two conditions that  $|x_m|^2 = 1$  and the DFT have a constant magnitude, by making  $\{x_m\}$  a maximal-length shift register sequence (this will be shown in Chapter 12). The result is called *direct-sequence spread spectrum*. Spreading codes can also be used to design orthogonal multipulse signal sets (see Problem 8-11). Unlike the pulse sets designed in Chapter 6, these can overlap one another completely in frequency.

Having established a method of designing a broadband  $h(t)$  that has very narrow autocorrelation  $\rho_h(t)$ , the next question is the effect of channel dispersion. Assuming the channel has impulse response  $b(t)$ , the output of the matched filter becomes

$$h(t) * b(t) * h^*(-t) = \rho_h(t) * b(t). \quad (8.118)$$

As before, the phase spectrum of  $h(t)$  does not matter. The non-ideal channel increases the time duration of the matched filter output. However, since  $\rho_h(t)$  can be kept very narrow when  $B$  is large,  $\rho_h(t) * b(t)$  will have time duration approximately equal to the duration of  $b(t)$ . As long as the duration of  $b(t)$  is smaller than the symbol interval  $T$ , the channel dispersion will not have a significant effect.

**Example 8-15.**

Spread spectrum is often used on radio channels, which suffer from multipath distortion (Chapter 5). Suppose we take a two-path model,

$$b(t) = \delta(t) + \alpha \cdot \delta(t - \tau), \quad (8.119)$$

where  $\tau$  is the relative delay of the second path. For the  $\rho_h(t)$  of Example 8-12, the isolated pulse output of the matched filter with this dispersive channel will be

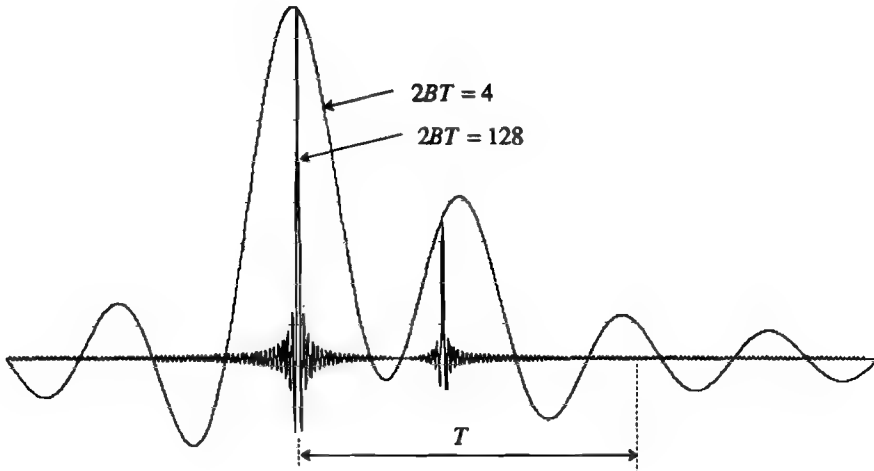
$$f(t) = b(t) * \rho_h(t) = \text{sinc}(2\pi Bt) + \alpha \cdot \text{sinc}(2\pi B(t - \tau)). \quad (8.120)$$

The symbol-rate samples of this isolated pulse are then

$$f(kT) = \delta_k + \alpha \cdot \text{sinc}(2\pi B(kT - \tau)). \quad (8.121)$$

As  $B$  gets large, assuming that  $|\tau| < T$ , the ISI gets small. This is illustrated in Figure 8-5. For  $2BT = 4$  or 300% excess bandwidth, the multipath distortion sampled at the symbol interval can still be fairly large at the output of the matched filter. For  $2BT = 128$ , even a





**Figure 8-5.** The isolated pulses at the output of a matched filter for a two-path multipath channel with  $\tau = 0.4 \cdot T$  and  $\alpha = 0.5$ . Two bandwidth expansions are shown:  $2BT = 4$  and  $2BT = 128$ . Note that the ISI will be small for  $2BT = 128$  as long as delay spread  $\tau$  is a little less than the symbol interval  $T$ , but for  $2BT = 4$  ISI will be significant even for very small  $\tau$ .

half-symbol multipath spread results in a very small ISI, because the basic pulse shape at the matched filter output decays so rapidly.  $\square$

This example illustrates the desirable effect of bandwidth expansion in terms of minimizing the effect of channel dispersion on the isolated pulse at the output of the matched filter. However, two caveats are in order:

- Spread spectrum with large  $2BT$  is not *immune* to ISI; for example, consider what happens when  $\tau = T$  in Example 8-15. In fact, if the multipath delay spread is  $\tau = T$ , ISI will be a problem no matter how large  $2BT$  may be! Spread spectrum successfully mitigates channel dispersion only where the time-delay spread is smaller than a symbol interval.
- Figure 8-5 illustrates that the timing recovery must be increasingly accurate as  $2BT$  increases. Fortunately, the broader bandwidth of the signal is helpful in increasing the timing recovery accuracy as well.

### Spread Spectrum and Jamming or Interference

We have shown that any bandwidth increase has no impact on the probability of error on white Gaussian noise channels, but it does help mitigate the effects of ISI. There are several other benefits to increasing bandwidth, including countering jamming and interference in military communications, hiding the communications from intruders, coexisting in the same spectrum with other uses, and multiple access (separating users sharing a common communications medium, Section 6.9 and Chapter 16). To illustrate these other advantages, we will consider one of them: immunity to jamming or broadband interference.

Assume that the noise on the channel is generated by a jammer. *Jamming* is a deliberate attempt to disrupt communication by generating a broadband interference signal. In practice, the jammer is limited in the power it can generate. The jammer generates bandlimited white noise with power  $P_J$  over the signal bandwidth  $B$ , with spectral density  $N_0 = P_J/2B$ . Since the jamming signal is white over the signal bandwidth, from the perspective of the receiver the error probability is the same as for white Gaussian noise (the presence or absence of out-of-band noise will be inconsequential). The receive SNR is now

$$SNR = \frac{P_S}{2BN_0} = \frac{P_S}{P_J}, \quad (8.122)$$

independent of bandwidth. The fact that the SNR is bandwidth-independent has profound implications, since from (8.111),  $P_e$  is now strongly dependent on the dimensionality  $2BT$ . In fact, as  $2BT$  increases by expanding  $B$ ,  $P_e$  decreases. For this reason,  $2BT$  is called the *processing gain*.

#### Example 8-16.

If the processing gain is  $2BT = 10^3$  (30 dB), the jammer power is effectively suppressed by 30 dB. That is, in going from  $2BT \approx 1$  to  $2BT = 10^3$  by expanding the bandwidth by 1000, 30 dB greater jammer power  $P_J$  can be tolerated with the same error probability.  $\square$

The processing gain can also be interpreted in signal space. The pulse  $h(t)$  defines a one-dimensional subspace, and our assumption is that the jammer does not know the direction of this subspace. Thus, the jammer must spread its power  $P_J$  evenly over all  $2BT$  dimensions of the subspace of signals bandlimited to  $B$  Hz and time limited to a symbol interval  $T$  (this is equivalent to the jammer transmitting bandlimited white noise). The matched filter responds to the jammer noise in the direction of the signal only, and hence at the output of the matched filter the jammer power is reduced by  $2BT$ .

The foregoing presumes that the jammer cooperates, and transmits bandlimited white noise, or equivalently spreads its power evenly over all dimensions of the signal subspace. But clearly the jammer would do better to concentrate its power in the direction of the signal, because then there would be no processing gain! Increasing the signal bandwidth is beneficial only if the jammer does not know the direction of the signal, and therefore must spread its jamming power equally in all directions. If the jammer transmits a one-dimensional signal, the jammer power in the direction of signal vector  $h(t)$  can fall anywhere between 100% (no processing gain) and 0% (infinite processing gain).

The use of the term "jammer" implies a military connotation, but in commercial microwave radio systems an important consideration is co-channel interference (Section 5.4). This interference, for example between two satellites within the aperture of a single antenna, between a satellite and terrestrial radio system, or among different users of a terrestrial cellular radio system, has similar characteristics to jamming. If the signal and interferer both spread their bandwidth, keeping their total powers the same, then a processing gain results. In fact, if we take steps to actively reduce

interference, the interferer can avoid transmitting in the one-dimension of the signal, resulting in an infinite processing gain! This is the principle behind the use of spread spectrum as a multiple access technique, as described in Section 6.9 and Chapter 16.

## 8.7. CAPACITY AND MODULATION

There are a number of ways of comparing different modulation techniques. The appropriate method depends on the context. Some of the measures of interest when making comparisons include:

- The *average transmitted power* is limited on most media, due to physical constraints or regulatory fiat. On some media, *peak transmitted power* is also of concern. Either of these limitations, together with the attenuation of the medium, will limit the received signal power. The *noise immunity* of the modulation system is measured by the minimum received power relative to the noise power required to achieve a given error probability.
- The *probability of error* is normally the basic measure of the fidelity of the information transmission. Of primary concern in some contexts is the probability of bit error, and in other contexts the probability of block error, for some appropriate block size.
- The *spectral efficiency* is of great concern on bandwidth-constrained media, such as radio. The spectral efficiency is the ratio of two parameters: the information bit rate available to the user and the bandwidth required on the channel.
- A measure that has not been discussed yet is the potential advantage of using coding techniques (Chapters 13 and 14). This advantage is different for modulation formats that are otherwise comparable. *Coding gain* is usually defined as the decrease in received signal power that could be accommodated at the same error probability if coding were used in conjunction with the modulation system.
- An important criterion in practice, although one we do not emphasize in this book, is the *implementation cost* or *design effort* required to realize a given modulation system.

The variety of measures for comparison of modulation systems makes it difficult to define one "standard measure of comparison". For example, for two modulation techniques, we can set the transmitted powers equal and compare the uncoded probability of error. However, the usefulness of this comparison will be compromised if the two modulation systems have different bandwidth requirements or provide different information bit rates for the same bandwidth.

In this section, we first illustrate how to make simple comparisons of uncoded modulation systems, addressing specifically baseband and passband PAM. Following this, a more sophisticated approach (based on a "rate-normalized SNR") is developed. This approach allows modulation systems to be compared against the fundamental limits of information theory (Chapter 4) and against one another in a way that is independent of bit rate or bandwidth requirements. The rate-normalized SNR takes

into account most of the parameters mentioned — transmit power, noise power, and spectral efficiency — and summarizes them in a single universal error probability plot that further displays the available coding gain.

Comparisons will be made under the following assumptions:

- The channel is an ideal bandlimited channel with bandwidth  $B$  and additive white Gaussian noise with power spectral density  $N_0$ .
- The average transmit power is constrained to be  $P_S$ . There is no peak power limitation.
- Symbol error probability adequately reflects the performance of the system. Moreover, the union bound is sufficiently accurate as an approximation to error probability.

### 8.7.1. Error Probability of PAM

The simplest approach to comparing modulation systems is to calculate their error probability as a function of all the relevant parameters. A convenient approximate formula for the probability of symbol error (based on the union bound) is given by (8.111), which we repeat here,

$$P_e \approx K \cdot Q(\sqrt{2BT \cdot \eta_A \cdot SNR}) . \quad (8.123)$$

This formula applies to any PAM system, as long as the effect of ISI is ignored, and expresses  $P_e$  in terms of the received  $SNR$ , dimensionality  $2BT$ , and a parameter of the signal constellation,  $\eta_A$ . Often it is desired to express the probability in terms of the spectral efficiency, which is

$$\nu = \frac{\log_2 M}{BT} , \quad (8.124)$$

where  $M$  is the number of points in the signal constellation.

It is instructive to determine  $\eta_A$  for two standard constellation designs.

#### Example 8-17.

For a baseband PAM constellation with  $M$  equally spaced points, let  $M$  be even and let the constellation consist of odd integers. Thus,  $\Omega_A = \{(2m - 1), 1 - M/2 \leq m \leq M/2\}$ , and the minimum distance is  $a_{\min} = 2$ . Assuming all the points in the constellation are equally probable, the variance is (calculating the average-squared value of only the positive points, taking advantage of symmetry)

$$\sigma_A^2 = \frac{2}{M} \sum_{m=1}^{M/2} (2m - 1)^2 = \frac{M^2 - 1}{3} . \quad (8.125)$$

Substituting into (8.112), and using (8.124),

$$\eta_A = \frac{3}{M^2 - 1} = \frac{3}{2^{2BT\nu} - 1} . \quad (8.126)$$

□

**Example 8-18.**

For a passband PAM  $L \times L$  square constellation with  $M = L^2$  points, and again using odd integers on each axis, the minimum distance is again  $a_{\min} = 2$ , and the variance for equally probable points can be shown to be  $\sigma_A^2 = 2(M - 1)/3$ . Substituting into (8.112) and (8.124),

$$\eta_A = \frac{3}{2(M - 1)} = \frac{3}{2(2^{BT^v} - 1)}. \quad (8.127)$$

□

In both the baseband and passband cases, as the number of points in the constellation increases, or equivalently the spectral efficiency increases,  $\eta_A$  gets smaller and as expected, from (8.123) we must increase  $SNR$  to hold  $P_e$  fixed.

It is useful to determine  $P_e$  when the highest possible symbol rate consistent with the Nyquist criterion is used, since this will maximize the  $SNR$  and minimize  $P_e$ . For the baseband case, the maximum symbol rate is  $1/T = 2B$ , and hence  $2BT = 1$ . For the passband case, a passband channel with bandwidth  $B$  corresponds to a complex baseband channel with bandwidth  $B/2$ , and thus the maximum symbol rate is  $1/T = B$  or  $2BT = 2$ . Expressed in terms of  $SNR$  and  $v$ , the resulting  $P_e$  is the same for both cases,

$$P_e \approx K \cdot Q \left[ \sqrt{3 \cdot \frac{SNR}{2^v - 1}} \right]. \quad (8.128)$$

This is a "universal" formula that applies to both baseband and passband PAM systems with square QAM constellations and the maximum possible symbol rate.

Using (8.123), the error probability a variety of baseband and passband constellations can be accurately estimated and compared, not just square QAM. A typical comparison would set the arguments of  $Q(\cdot)$  equal for two constellations, to force their  $P_e$ 's to be approximately equivalent, and then solve for the relative  $SNR$ 's.

**Example 8-19.**

Both a baseband binary antipodal constellation and a passband square 4-QAM constellation with the maximum feasible symbol rate obey (8.128). They also have the same spectral efficiency, 2 bits/sec-Hz. Thus, these two systems will have, to accurate approximation, the same  $P_e$  if their  $SNR$ 's are the same. (Taking into account the error coefficients,  $K = 1$  for binary antipodal and  $K = 2$  for 4-PSK, so in actuality the binary antipodal constellation will have half the error rate.) Intuitively, this can be interpreted as follows. For the same minimum-distance (and hence  $P_e$ ), the passband constellation requires 3 dB greater transmit power, since the radius of the constellation points will be  $\sqrt{2}$  larger. However, the passband bandwidth is also twice as great, for the same symbol rate, allowing in 3 dB more noise. Thus, at a fixed  $P_e$  the net passband  $SNR$  is the same as in the baseband case, since both the signal and the noise are 3 dB larger. □

The comparison in Example 8-19 is straightforward because the two modulation systems being compared have the same spectral efficiency (2 bit/sec-Hz). The passband system requires twice the bandwidth, but also has twice as many information bits per symbol. However, if two systems with different spectral efficiencies are compared, things get more complicated.

**Example 8-20.**

If we use the same two constellations as in Example 8-19, but make them both passband, we are comparing 2-PSK (binary antipodal) against 4-PSK (square 4-QAM). For a 2-PSK passband constellation,  $2BT = 2$  and  $\eta_A = 1$ , and thus  $P_e \approx Q(\sqrt{2 \cdot SNR})$ . Setting the arguments of  $Q(\cdot)$  equal,

$$2 \cdot SNR_{2-PSK} = 1 \cdot \frac{3 \cdot SNR_{4-PSK}}{2^2 - 2}, \quad (8.129)$$

or  $2 \cdot SNR_{2-PSK} = SNR_{4-PSK}$ . We conclude that 4-PSK requires a 3 dB higher  $SNR$  for the same error probability. (Again, this ignores the effect of the error coefficient, which will be  $K = 1$  for 2-PSK and  $K = 2$  for 4-PSK.) The bandwidth requirement, and hence noise powers, are the same for both systems. In order to maintain the same minimum-distance, the transmitted power for 4-PSK has to be 3 dB higher. Since the noise is the same, the  $SNR$  also has to be 3 dB larger.  $\square$

Example 8-20 indicates that 2-PSK is "better" than 4-PSK, in the sense that at the same symbol rate it operates at the same approximate error probability with a 3 dB lower  $SNR$ . This could be misleading, however, because for the same symbol rate, the 2-PSK system is achieving only half the spectral efficiency (1 bit/sec-Hz vs. 2 bit/sec-Hz). In order to achieve the same bit rates for 2-PSK and 4-PSK, the symbol rate of the 2-PSK system would have to be twice as large, which would require twice the channel bandwidth and increase the noise by 3 dB. Thus, a 2-PSK system and a 4-PSK system operating at the same bit rate would require the same  $SNR$  to achieve the same  $P_e$ .

The complications and subtleties of comparing modulation systems that have different bit rates or spectral efficiencies, as in Example 8-20, demonstrate that a better approach is needed. The universal formula for  $P_e$  in (8.128) gives a hint as to a better approach, since it shows that, when expressed in terms of spectral efficiency, all baseband and passband square QAM constellations are equivalent. Another simplification of this formula is that  $P_e$  does not depend on  $SNR$  or  $v$  individually, but only through the ratio  $SNR/(2^v - 1)$ . It is helpful, therefore, to define a new parameter  $SNR_{\text{norm}}$ , which is called the *rate-normalized SNR*, as

$$SNR_{\text{norm}} = \frac{SNR}{2^v - 1}. \quad (8.130)$$

Now, the  $P_e$  is a function of only two parameters,  $K$  and  $SNR_{\text{norm}}$ ,

$$P_e \approx K \cdot Q(\sqrt{3 \cdot SNR_{\text{norm}}}). \quad (8.131)$$

Two square baseband or passband QAM constellations with the maximum symbol rate and the same  $SNR_{\text{norm}}$  will have approximately the same  $P_e$ .

The utility of (8.131) is that it expresses very succinctly  $P_e$  for a variety of PAM systems, including baseband, passband, and different bit rates and symbol rates. The simplicity of this result leads us to speculate that there may be something fundamental about this tradeoff between  $SNR$  and  $v$  expressed in  $SNR_{\text{norm}}$ . Indeed there is, although we will have to take a diversion, calculating the capacity of the channel, to uncover it.

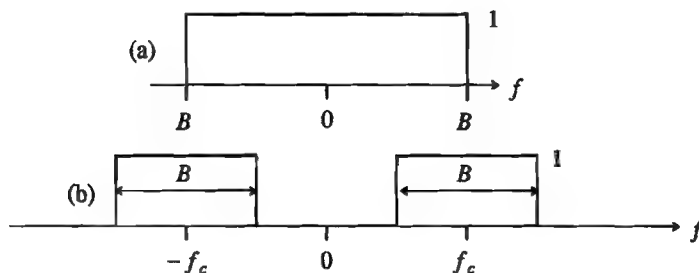
### 8.7.2. Capacity of the Ideal Gaussian Channel

Two approaches to comparing modulation systems operating over the same channel are first, to compare them directly, or second, to compare them against the fundamental limits of channel capacity (Chapter 4). The channel capacity tells us the maximum bit rate (or equivalently spectral efficiency) that can be obtained on the underlying channel. Comparing against capacity has a couple of benefits. First, it gives an indirect comparison of the systems against one another. Second, the comparison against fundamental limits gives us a valuable benchmark, because it indicates the maximum possible benefits of doing channel error-correction coding (Chapters 13 and 14).

The capacity of an ideal bandlimited Gaussian channel with additive white Gaussian noise will now be derived, and subsequently compared against the spectral efficiency of several modulation techniques operating over this same channel. A general way to compare a given modulation system against this capacity, based on the rate-normalized SNR already encountered in QAM modulation, will be uncovered.

The frequency response of an ideal channel with bandwidth  $B$  Hz is shown in Figure 8-6 for two cases, baseband and passband. The convention is that the bandwidth of the channel is  $B$  in both cases. This implies that the baseband channel is equivalent to the passband channel for the specific carrier frequency  $f_c = B/2$ . Intuitively, we would not expect the carrier frequency to affect the capacity of the channel, since the noise is white, and thus we expect the capacity of the two channels in Figure 8-6 to be identical. In fact, that is the case.

We are interested in calculating the capacity  $C$  of these two channels, where capacity has the units bits/sec. Thus,  $C$  can be directly compared to the bit rate achieved by a given modulation system. The capacity is calculated under a transmitted power constraint, so that the transmitted signal is constrained to have power  $P_S$ . Also of interest is the spectral efficiency  $\nu$ , which has the units of bits/sec-Hz. We define  $\nu_c$  as the spectral efficiency of a system operating at the limits of capacity, and thus



**Figure 8-6.** An ideal bandlimited channel with bandwidth  $B$ . (a) The baseband case, and (b) the passband case.

$$v_c = C/B. \quad (8.132)$$

The channel coding theorem (Chapter 4) says that a certain spectral efficiency  $v_c$  can be achieved with transmit power  $P_S$  in the sense that an arbitrarily small probability of error can be achieved by *some* modulation and coding scheme. Further, it says that if you try to achieve a higher  $v$  at this  $P_S$ , the probability of error is necessarily bounded away from zero for *all* modulation and coding schemes. The tradeoff between  $v_c$  and  $P_S$  as quantified by the channel capacity theorem is thus a fundamental limit against which all modulation systems can be compared.

The capacity of the ideal channels in Figure 8-6 with additive white Gaussian noise is simple to calculate using the capacity of a vector Gaussian channel (Chapter 4), together with the results of Chapter 7. We will do the calculation for the baseband case since it is slightly easier, although the passband case is also straightforward (Problem 8-14). Utilizing the Landau-Pollak theorem (Section 7.4) and the orthogonal expansion of the signal subspace of Section 7.1, any transmitted signal with bandwidth  $B$  Hz can be approximately represented in a time interval of length  $T$  by  $2BT$  orthonormal waveforms, with increasing accuracy as  $T \rightarrow \infty$ . From Section 8.4, the minimum-distance receiver will generate a set of  $2BT$  decision variables, by (8.83),

$$\mathbf{C} = \mathbf{S} + \mathbf{N}, \quad (8.133)$$

where  $\mathbf{S}$  is the  $2BT$ -dimensional vector of signal components, and  $\mathbf{N}$  is a vector of Gaussian independent noise components, each component having variance  $\sigma^2 = N_0$ . In this case, since the signal, noise, and channel are real-valued, all vectors in (8.133) are real-valued, and we use  $\mathbf{N}$  for the noise vector rather than  $\mathbf{Z}$ .

The capacity of channel (8.133), consisting of an  $N$ -dimensional real-valued vector signal in real-valued vector Gaussian noise, with total signal variance  $\sigma_x^2$  and noise variance per dimension  $\sigma^2$ , is given by (4.36),

$$C_{VG} = \frac{N}{2} \log_2(1 + SNR), \quad SNR = \frac{\sigma_x^2}{N\sigma^2}. \quad (8.134)$$

This is the capacity for a *single use* of the vector channel, or equivalently the capacity for the continuous-time channel over a time interval of length  $T$ . The signal-to-noise ratio  $SNR$  is defined as the ratio of the total signal variance to the total noise variance.

The constraint that the transmitted power is  $P_S$  implies that the average transmitted energy in time interval  $T$  must be  $T P_S$ , and thus

$$E\left[\int_0^T S^2(t) dt\right] = \sum_{n=1}^N E[S_n^2] = \sigma_x^2 = T P_S. \quad (8.135)$$

Defining  $C_T$  as the capacity for time interval  $T$ , with this power constraint,

$$C_T = BT \cdot \log_2(1 + SNR), \quad SNR = \frac{T P_S}{2BTN_0} = \frac{P_S}{2N_0B} \text{ bits}. \quad (8.136)$$

In this case,  $SNR$  can again be interpreted as signal-to-noise ratio, since the numerator  $P_S$  is the total signal power at the channel output, and the denominator is the total



noise power within the signal bandwidth (the noise spectral density  $N_0$  times the total bandwidth, which is  $2B$  for positive and negative frequencies).

The capacity per unit time is

$$C = C_T/T = B \cdot \log_2(1 + SNR) \text{ bits/sec.} \quad (8.137)$$

This expression for the capacity of a bandlimited channel is known as the *Shannon limit*. Alternative proofs and interpretation of this result are given in [7,8].

### Fundamental Limit In Spectral Efficiency

The spectral efficiency is the bit rate per unit time (capacity) divided by the bandwidth, and thus the maximum spectral efficiency predicted by the channel capacity is

$$\nu_c = C/B = \log_2(1 + SNR) \text{ bits/sec-Hz.} \quad (8.138)$$

If  $\nu$  is the spectral efficiency of any practical modulation scheme operating at signal-to-noise ratio  $SNR$ , then we must have  $\nu \leq \nu_c$ .

### Rate-Normalized Signal-to-Noise Ratio

Rewriting (8.138) in a different way, if a modulation system is operating at the limits of capacity with signal-to-noise ratio  $SNR$  and spectral efficiency  $\nu_c$ , then

$$\frac{SNR}{2^{\nu_c} - 1} = 1. \quad (8.139)$$

This relation has a striking similarity to  $SNR_{\text{norm}}$  defined in (8.130), and  $SNR_{\text{norm}}$  was shown in (8.131) to largely determine  $P_e$  for a rectangular baseband or passband QAM constellation. The only difference is that  $\nu$ , the spectral efficiency of the PAM modulator, is substituted for  $\nu_c$ , the spectral efficiency at capacity limits. The combination of (8.131) and (8.139) suggests that  $SNR_{\text{norm}}$  is a fundamental and useful parameter of a modulation system [9]. In fact, since  $\nu \leq \nu_c$  for a system operating short of the capacity limit,

$$SNR_{\text{norm}} = \frac{2^{\nu_c} - 1}{2^{\nu} - 1} \geq 1. \quad (8.140)$$

This is another way of expressing the Shannon limit on the operation of a given modulation system; if the modulation system operates at signal-to-noise ratio  $SNR$  with spectral efficiency  $\nu$ , and the corresponding  $SNR_{\text{norm}} > 1$ , then there is nothing fundamental preventing that system from having an arbitrarily small  $P_e$ . (If it has a large  $P_e$ , that is only because it is falling short of fundamental limits). Conversely, if  $SNR_{\text{norm}} < 1$ , the  $P_e$  of the system is necessarily bounded away from zero, because the parameters of the system ( $SNR$  and  $\nu$ ) are violating Shannon limits. In this case, the capacity theorem does not prevent  $P_e$  from being small, but it does guarantee that there is nothing we could do (like adding error-control coding) to make  $P_e$  arbitrarily small, short of changing the parameters  $SNR$  and/or  $\nu$ . Thus,  $SNR_{\text{norm}} > 1$  is the region where we want to operate on an ideal bandlimited white Gaussian noise channel.

It is useful to plot the relationship between  $SNR$  and  $SNR_{\text{norm}}$ , where both are expressed in dB, as in Figure 8-7. Taking the logarithm of (8.130),

$$SNR_{\text{norm,dB}} = SNR_{\text{dB}} - \Delta SNR_{\text{dB}}, \quad \Delta SNR_{\text{dB}} = 10 \cdot \log_{10} (2^v - 1). \quad (8.141)$$

At large spectral efficiencies, the unity term can be ignored, and  $\Delta SNR_{\text{dB}}$  approaches an asymptote of  $\Delta SNR_{\text{dB}} \approx 3v$ . Thus, for a hypothetical high spectral efficiency system operating at the limits of capacity, 3 dB of additional  $SNR$  is required to increase spectral efficiency by one bit/sec-Hz. At low spectral efficiencies, a larger increase in  $SNR$  is required. Remarkably, PAM systems with square QAM constellations operating at a constant  $P_e > 0$  obey exactly the same tradeoff between  $v$  and  $SNR$ , as indicated by (8.131). (Although the tradeoff is the same, they will require a higher absolute  $SNR$  to achieve a reasonable  $P_e$ , as will be seen shortly.)

For any modulation system, the gap (usually expressed in dB) between  $SNR_{\text{norm}}$  and unity (the minimum value of  $SNR_{\text{norm}}$ ) is a measure of how far short of fundamental limits the modulation scheme falls. Specifically, it is a measure of how much the transmitted power (or equivalently the  $SNR$ ) must be increased to achieve a given spectral efficiency, relative to the lower bound on transmitted power (or  $SNR$ ) predicted by capacity. The usefulness of  $SNR_{\text{norm}}$  is that it summarizes  $SNR$  and  $v$  in a single parameter, and the Shannon limit is very simply expressed in terms of  $SNR_{\text{norm}}$ .

### 8.7.3. Using Normalized SNR in Comparisons

While  $SNR_{\text{norm}} = 1$  corresponds to a hypothetical system operating at capacity, all practical modulation schemes such as those considered in Chapters 6 and 7 will have a non-zero error probability for all values of  $SNR_{\text{norm}}$ . A useful way to characterize  $P_e$  is to parameterize it on  $SNR_{\text{norm}}$ , because  $SNR_{\text{norm}}$  expresses both  $SNR$  and

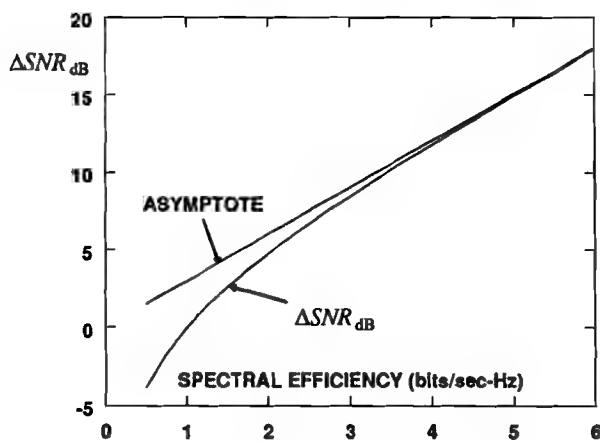


Figure 8-7. The difference between  $SNR$  and  $SNR_{\text{norm}}$  in dB plotted against spectral efficiency. The "asymptote" is the same relationship ignoring the "1" term.

$v$  in a single parameter, and because the Shannon limit is so simply characterized in terms of  $SNR_{\text{norm}}$ . In this section we illustrate how the symbol error probability  $P_e$  can be calculated as a function of  $SNR_{\text{norm}}$  for different modulation techniques. This relationship gives us several types of information:

- Comparisons can be made between different modulation techniques. For two modulation systems operating at the same  $P_e$  (as approximated by the union bound, and ignoring the effect of the error coefficient  $K$ ), and at the same spectral efficiency, if the superior modulation system allows  $SNR_{\text{norm}}$  to be 3 dB lower, then it allows 3 dB lower transmit power. Alternatively, if the two systems are operating at the same  $SNR$ , then the superior system will operate at a spectral efficiency that is one bit/sec-Hz higher (asymptotically at high  $v$ ).
- Comparisons can be made between a modulation system and fundamental limits. At a given  $P_e$  and  $v$ , the difference between  $SNR_{\text{norm}}$  and unity (usually expressed in dB) tells us how far the modulation system is operating from fundamental limits, in the sense that it is requiring a higher  $SNR$  or lower  $v$  to achieve the same spectral efficiency. This quantifies, for example, the ultimate potential benefit of adding error-correction coding to the system (Chapters 13 and 14).
- Reasonable comparisons between modulation systems operating at different information bit rates and spectral efficiencies can be made. As we saw in Example 8-20, such a comparison can be conceptually difficult, and yet is of practical interest. For example, we might want to compare two schemes utilizing the same bandwidth but having a different number of points in the constellation (and hence different spectral efficiency). Comparing them each against the Shannon limit is an indirect way of comparing them against each other.

We are interested in a wide range of error probabilities (some applications are more demanding than others), and thus it is useful to plot the functional relationship between  $P_e$  and  $SNR_{\text{norm}}$ , and compare to capacity ( $SNR_{\text{norm}} = 1$ ). This will now be illustrated for several modulation systems.

### Baseband and Passband PAM

Earlier in this section,  $P_e$  was estimated for both baseband and passband QAM constellations. The result was (8.123), where the parameter  $\eta_A$  is given by (8.126) (baseband case) and (8.127) (passband case). Expressing  $P_e$  in terms of  $SNR_{\text{norm}}$ , rather than  $SNR$ , (8.123) can be rewritten as

$$P_e \approx K \cdot Q(\sqrt{\gamma_A \cdot SNR_{\text{norm}}}), \quad (8.142)$$

where

$$\gamma_A = \eta_A (2^v - 1). \quad (8.143)$$

This assumes that the bandwidth on the channel is the minimum consistent with the Nyquist criterion. For the baseband case, from (8.126),

$$\gamma_A = \frac{a_{\min}^2 (2^v - 1)}{4\sigma_A^2}, \quad (8.144)$$

and for the passband case, from (8.127),

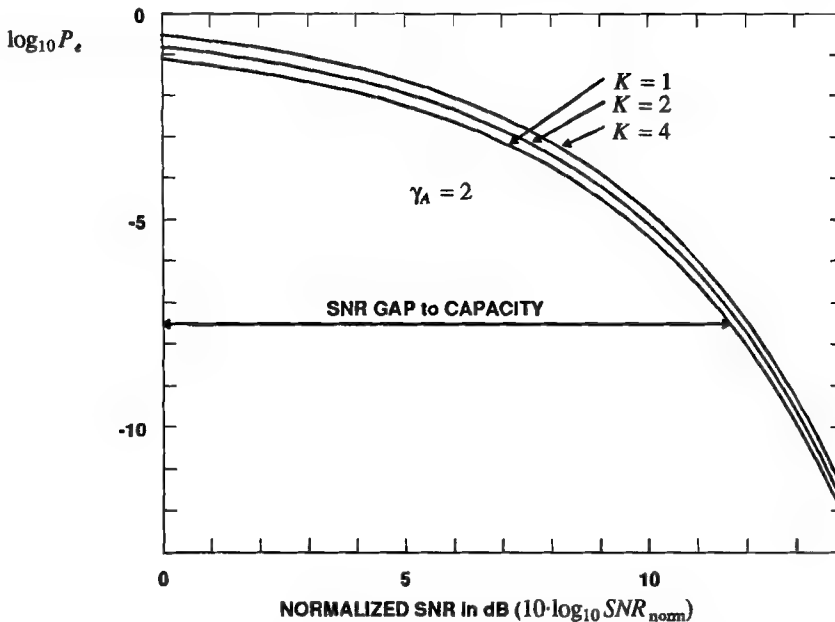
$$\gamma_A = \frac{a_{\min}^2 (2^v - 1)}{2\sigma_A^2}. \quad (8.145)$$

**Example 8-21.**

For square QAM constellations, as shown in (8.131), the remarkably simple result is that  $\gamma_A = 3$ . This holds for all cases where the number of points in the constellation is even (baseband case) or the square of an even number (passband case).  $\square$

For other PAM constellations,  $\gamma_A$  is a parameter of the constellation that is independent of scaling, but is a function of the geometry of the constellation. Remarkably, the  $P_e$  of any PAM constellation across a wide range of SNR's, is accurately summarized in this single parameter  $\gamma_A$ . It can be determined directly from (8.144) or (8.145). The error coefficient  $K$  is also relevant, although much less important.

We can plot  $P_e$  vs  $SNR_{\text{norm}}$  under different conditions, and get a set of universal rate-normalized curves. First, in Figure 8-8,  $\gamma_A$  is held fixed (at  $\gamma_A = 2$ ), and  $K$  is



**Figure 8-8.** A plot of  $P_e$  vs.  $SNR_{\text{norm}}$  for passband PAM assuming  $\gamma_A = 2$  and three typical values of  $K$ . This illustrates that  $K$  has a relatively minor effect on the error probability.

varied. This set of curves has several interesting interpretations. First, it shows how large an  $SNR_{\text{norm}}$  is required to achieve a given error probability for these assumed parameters. As expected, the required  $SNR_{\text{norm}}$  increases as  $P_e$  gets smaller. The Shannon limit dictates that  $SNR_{\text{norm}} > 1$ , or  $10 \cdot \log_{10} SNR_{\text{norm}} > 0$ . Since the channel capacity theorem guarantees the feasibility of achieving any (arbitrarily small) error probability, it is theoretically possible to achieve any point on the 0 dB  $SNR_{\text{norm}}$  axis; conversely, since  $SNR_{\text{norm}} \geq 1$ , the probability of error will be theoretically bounded away from zero at any point to the left of the 0 dB  $SNR_{\text{norm}}$  axis. In this sense, the 0 dB axis represents the limit on deliverable performance as dictated by Shannon limit. At a given  $P_e$  the horizontal distance between the 0 dB  $SNR_{\text{norm}}$  axis and the curve, labeled "SNR GAP to CAPACITY", represents the increase in  $SNR_{\text{norm}}$  required relative to capacity. Also, the horizontal distance between two curves represents the difference in  $SNR_{\text{norm}}$  required for two different signal constellations to achieve the same  $P_e$ . This gap can be made up in one of two ways: operate the system at a higher SNR, or at a lower  $v$ .

By definition, the SNR gap to capacity goes to zero as  $SNR_{\text{norm}} \rightarrow 1$ . What may be surprising is that  $P_e$  can be small (like  $10^{-1}$ ) at this crossover point, or even for  $SNR_{\text{norm}} < 1$ . Doesn't the channel capacity theorem rule out any useful operation for  $SNR_{\text{norm}} < 1$ ? Two points should be made about this behavior. First, since the error probability is based on the union bound, it is generally not wise to trust these quantitative results at low SNR (high  $P_e$ ), except for modulation schemes for which the union bound is exact (such as binary antipodal signaling). Second, although it would be tempting to assert that the channel capacity tells us something specific about the error probability of any modulation scheme operating at  $SNR_{\text{norm}} < 1$ , in fact it only asserts that in this region the error probability is bounded away from zero. It does not tell us what that bound is. Thus, the channel capacity theorem does not rule out any non-zero error probability at the point where  $SNR_{\text{norm}} = 1$ .

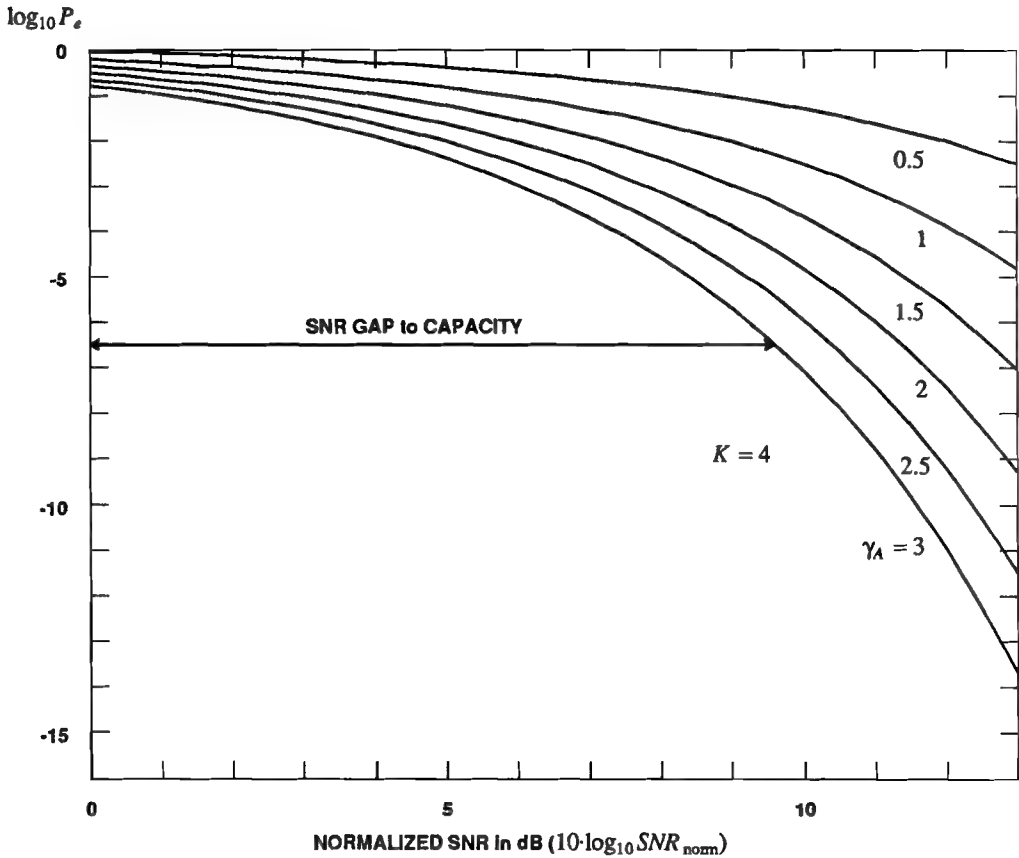
In Figure 8-8 the effect of  $K$  on  $SNR_{\text{norm}}$  is small, emphasizing that  $K$  has a relatively minor influence on  $P_e$ . The effect of  $\gamma_A$  is much more significant, as illustrated in Figure 8-9. The major factor distinguishing different signal constellations is  $\gamma_A$ . We will calculate  $\gamma_A$  for a couple of cases to illustrate this.

#### Example 8-22.

All rectangular QAM constellations are equivalent, in the sense that they require the same  $SNR_{\text{norm}}$  to achieve a given error probability. That tradeoff between  $SNR_{\text{norm}}$  and  $P_e$  is the  $\gamma_A = 3$  curve in Figure 8-9. For example, at an error rate of  $P_e = 10^{-6}$ , the SNR gap to capacity is about 9 dB, independent of the size of the constellation. However, at a fixed  $P_e$ , square QAM constellations do require different unnormalized SNRs, since for the passband case

$$SNR = SNR_{\text{norm}} \cdot (2^v - 1) = SNR_{\text{norm}} \cdot (M - 1). \quad (8.146)$$

As  $M$  increases, the SNR must increase in proportion to  $M - 1$  because of the need to increase the signal power to maintain the same minimum distance. Looking at it another way, as the spectral efficiency  $v$  increases, the SNR must be increased in proportion to  $(2^v - 1)$ .  $\square$



**Figure 8-9.** A plot of  $P_e$  vs.  $SNR_{\text{norm}}$  for passband PAM, assuming  $K = 4$  and different values of  $\gamma_A$ .

#### Example 8-23.

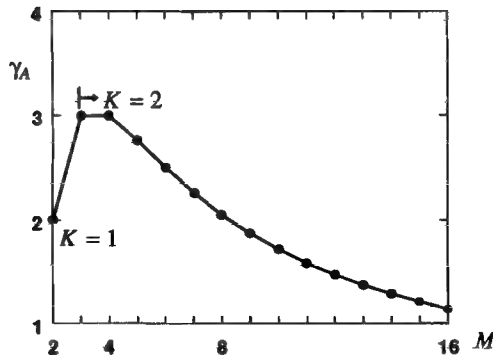
For a PSK signal constellation, all the points fall on the unit circle, and thus  $\sigma_A^2 = 1$  independent of the distribution of signal constellation points. It is straightforward to show that  $a_{\min} = 2 \sin(\pi/M)$ , and thus,

$$\gamma_A = 2(M-1) \sin^2\left(\frac{\pi}{M}\right). \quad (8.147)$$

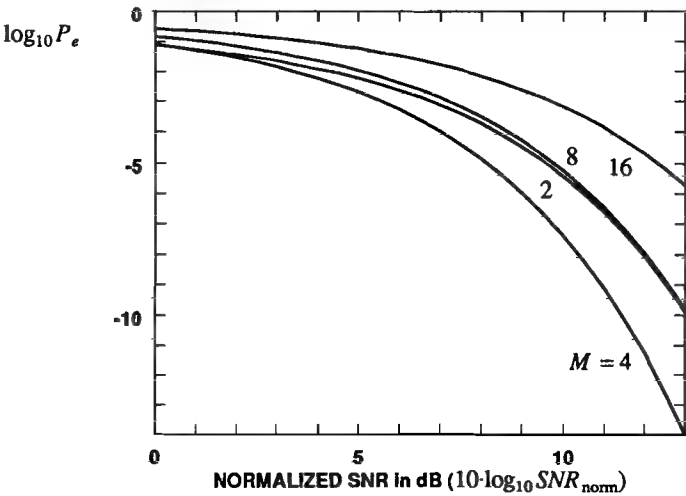
In this case,  $\gamma_A$  is strongly dependent on  $M$ , in contrast to rectangular QAM. This dependence is plotted in Figure 8-10, where the largest  $\gamma_A$  is 3, the same as rectangular QAM, for  $M = 3$  and  $M = 4$ . Thus, the SNR gap to capacity for PSK is the same for 3-PSK and 4-PSK as it is for rectangular QAM. The equivalence at  $M = 4$  is obvious, since 4-PSK is in fact a square QAM constellation. Both 2-PSK (binary antipodal) and  $M$ -PSK for  $M > 4$  are inferior to rectangular QAM in the sense that they require a larger  $SNR_{\text{norm}}$  to achieve the same  $P_e$  (higher  $SNR$  at the same  $v$  or lower  $v$  at the same  $SNR$ ). In the case of 2-PSK, which is equivalent to binary antipodal signaling, its gap is larger than QAM because it is a passband PAM system that fails to use the quadrature axis (we have shown previously that

a *baseband* PAM binary antipodal constellation has  $\gamma_A = 3$ ).

The SNR gap to capacity for PSK increases rapidly at a given  $P_e$  as  $M$  increases. Intuitively, this is because PSK does not efficiently pack the circularly-shaped constellation with a regular grid of points, and thus suffers in spectral efficiency as compared to square QAM constellations. We can also plot the  $P_e$  directly, as shown in Figure 8-11, for different  $M$ , taking into account the corresponding  $K$ .  $M = 4$  has the smallest gap (equivalent to rectangular QAM), and  $M = 2$  and 8 have roughly the same gap (because the  $\gamma_A$  is about the same, as seen in Figure 8-10). Choosing large values of  $M$  for PSK results in significantly poorer performance.  $\square$



**Figure 8-10.** The relationship of  $\gamma_A$  to  $M$  for the PSK constellation. The cases  $M = 3$  and  $M = 4$  have the smallest SNR gap to capacity.



**Figure 8-11.**  $P_e$  vs.  $SNR_{\text{norm}}$  for a PSK constellation and several values of  $M$ .

If it is desired to compare two constellations against one another analytically, the simplest approach is to set the arguments of  $Q(\cdot)$  to be equal, which ignores the effect of  $K$ . In other words, compare their  $\gamma_A$ .

**Example 8-24.**

To compare rectangular QAM against PSK, set

$$3 \cdot SNR_{\text{norm,QAM}} = 2(M-1) \cdot \sin^2\left(\frac{\pi}{M}\right) \cdot SNR_{\text{norm,PSK}} \quad (8.148)$$

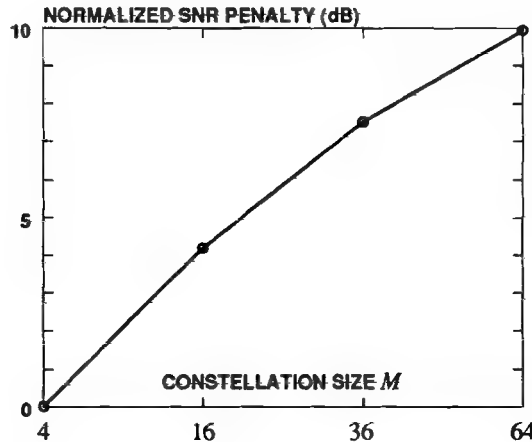
to yield

$$\frac{SNR_{\text{norm,PSK}}}{SNR_{\text{norm,QAM}}} = \frac{3}{2(M-1) \cdot \sin^2(\pi/M)} \quad (8.149)$$

This relationship is plotted vs.  $M$  in Figure 8-12 in dB. The penalty in  $SNR_{\text{norm}}$  for using PSK is shown as a function of  $M$ . For all  $M$  except four, PSK requires a higher SNR (by the amount shown) to achieve a similar error probability. All values of  $M$  are squares of even integers, which are the only square QAM constellation sizes.  $\square$

## Spread Spectrum

Our examples of the SNR gap to capacity for PAM thus far have presumed that the maximum feasible symbol rate in relation to channel bandwidth is used. In spread spectrum, a much lower symbol rate is used, and the SNR gap to capacity will expand accordingly. We can quantify this effect as follows. Considering the bandpass case, from substituting (8.127) into (8.123),



**Figure 8-12.** The penalty in dB for PSK in comparison to QAM, vs. constellation size  $M$ . The plot starts at  $M = 4$  because this is the smallest QAM constellation, and only the  $M$  that are perfect squares are shown. The horizontal axis is not to scale.



$$P_e \approx K \cdot Q(\sqrt{3 \gamma_{SS} SNR_{\text{norm}}}), \quad (8.150)$$

where the additional factor is

$$\gamma_{SS} = \frac{BT \cdot (2^v - 1)}{2^{BTv} - 1}. \quad (8.151)$$

Since  $v$  is a function of  $BT$ , it is useful to express  $\gamma_{SS}$  in terms of  $M$ , the number of points in the constellation,

$$\gamma_{SS} = \frac{BT \cdot (M^{1/BT} - 1)}{M - 1}. \quad (8.152)$$

For the maximum symbol rate,  $2BT = 2$  and  $\gamma_{SS} = 1$ . More generally, however,  $\gamma_{SS} < 1$ , forcing  $SNR_{\text{norm}}$  to be larger for the same  $P_e$  and increasing the SNR gap to capacity. This implies that coding is more beneficial in spread spectrum systems.

#### Exercise 8-2.

Show that as  $BT \rightarrow \infty$ ,  $\gamma_{SS} \rightarrow (\log_e M)/(M-1)$ .  $\square$

The effect of  $\gamma_{SS}$  is to increase the SNR gap to capacity. Asymptotically, the gap is increased by  $(M-1)/\log_e M$  for a signal constellation of size  $M$ . For  $M = 4$ , the smallest  $M$  for which the formula for  $\gamma_A$  is valid, the SNR gap to capacity is increased by  $3/\log_e 4 = 2.16$  (3.3 dB).

Penalizing spread spectrum in its SNR gap to capacity, although understandable in terms of its reduced spectral efficiency, is unfair when we realize that multiple spread spectrum signals can coexist within the same bandwidth, as in code-division multiple access described in Section 6.9 and Chapter 16. The increase in SNR gap to capacity calculated here is for a *single* spread spectrum signal occupying the bandwidth. Also, these results are for white noise with fixed power spectral density on the channel, which is not always the context in which spread spectrum is used.

### Orthogonal Multipulse

In orthogonal multipulse, we start with a set of  $N$  orthonormal signals  $\{\phi_n(t), 0 \leq n \leq N\}$ . If these signals are bandlimited to  $B$  Hz for transmission over the baseband channel, and if the dimensionality  $N$  is relatively large, then these pulses can be largely confined to an interval of length  $T = N/2B$ . Thus, over this interval the transmitted signal is  $\sigma_g \cdot \phi_m(t)$  for some  $m$ , and the energy is  $\sigma_g^2$ , or the power is

$$P = \frac{\sigma_g^2}{T} = \frac{2B \cdot \sigma_g^2}{N}. \quad (8.153)$$

As shown in Chapter 7, the spectral efficiency is

$$v = \frac{\log_2 N^2}{N}, \quad 2^v - 1 = N^{2/N} - 1. \quad (8.154)$$

The minimum-distance receiver again consists of a set of  $N$  correlators, and the resulting  $N$ -dimensional signal vector has a minimum distance of  $d_{\min} = \sqrt{2} \cdot \sigma_g$ , with

all  $N-1$  other signals at the minimum distance. Thus, the union bound gives

$$P_e \approx (N-1) \cdot Q\left(\sqrt{\frac{2 \cdot \sigma_g^2}{4N_0}}\right) = (N-1) \cdot Q\left(\sqrt{\gamma_N \cdot SNR_{\text{norm}}}\right), \quad (8.155)$$

where

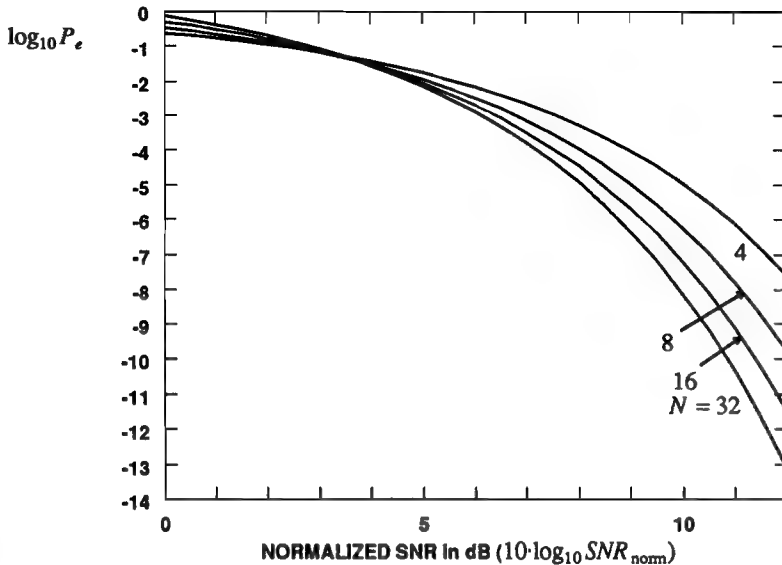
$$\gamma_N = \frac{1}{2} N (N^{2/N} - 1). \quad (8.156)$$

$\gamma_N$  is monotonically increasing in  $N$ , which reduces the error probability, but the multiplier  $K = N-1$  increases it. The resulting error probability is plotted for several values of  $N$  in Figure 8-13; the net effect is that the SNR gap to capacity decreases as  $N$  increases.

From Figure 8-13 it appears that orthogonal multipulse gives similar performance to QAM, in the sense that the SNR gap is similar. However, as we will see in Chapter 14, there are known ways to use channel coding to "shrink the gap" in the case of PAM, but there is no way to do this directly with orthogonal multipulse.

### Orthogonal Multipulse with PAM

Choosing the same set of orthonormal pulses as in orthogonal multipulse modulation, the transmitted signal will be of the form



**Figure 8-13.**  $P_e$  vs.  $SNR_{\text{norm}}$  for orthogonal multipulse for several values of  $N$ , where  $N$  is the dimensionality.

$$S(t) = \sum_{n=0}^{N-1} A_n \cdot \sigma_h \phi_n(t), \quad (8.157)$$

for a time interval of length  $T$ , bandwidth  $B/2$ , and  $N = BT$ . We assume that the  $N$  data symbols independent, identically distributed, each with variance  $\sigma_A^2$ , in which case the energy per symbol interval is  $N \sigma_A^2 \sigma_h^2$ , and the average transmitted power is

$$P = \frac{N \sigma_A^2 \sigma_h^2}{T} = B \sigma_A^2 \sigma_h^2. \quad (8.158)$$

This average transmitted power is the same as in PAM, for the same pulse energy and data-symbol constellations. As shown in Chapter 7, the spectral efficiency is also the same as PAM,  $v = \log_2 M$ .

At the receiver, we apply a set of  $N$  correlators, which separate out the  $N$  data symbols and apply them to a set of  $N$  identical slicers. The data symbol and noise at the slicer inputs are the same as in PAM. If we define  $P_e$  as the probability of error for a single such slicer (rather than the probability of one or more of the  $N$  data symbols being in error), then  $P_e$  is the same as in PAM.

In summary, since the spectral efficiency, transmit power, and  $P_e$  are all identical to PAM, the earlier results for PAM apply directly to orthogonal multipulse plus PAM, assuming the same data-symbol alphabet is used.

## 8.8. QUANTUM NOISE in OPTICAL SYSTEMS

In the preceding sections of this chapter we have considered only Gaussian noise. Fiber optics with intensity modulation and direct-detection receivers often operate in a thermal-noise limited regime, in which case these results apply directly. However, it is possible to reduce the thermal noise to insignificant levels using either optical amplifiers (Chapter 5) or coherent detection (considered here). In this case, we encounter a fundamental limit to the performance of optical systems that is due to the discrete nature of photons and the randomness of the number of photons that will arrive in any interval of time. This is known as the *quantum limit*. Optical systems operating in this quantum-limited regime must be analyzed differently than in the thermal-noise regime.

In recent years, optical fiber techniques have evolved rapidly. Further evolution is inevitable, so we concentrate on fundamentals rather than highly specific technologies. We begin with a discussion of direct detection receivers, since these are currently the only type in commercial use. We then consider coherent techniques, which have been explored in the laboratory.

### 8.8.1. Direct Detection Receivers

In direct detection systems for optical fiber (Section 5.3) the intensity or power of the light is modulated by a data signal, and the detector converts the received power into an electrical current. Since power is always non-negative, the data signal is non-negative. The simplest case is *on-off keying (OOK)*, where a "zero" is represented by

zero intensity and a "one" is represented by positive intensity. Virtually all commercial optical fiber digital transmission systems today use OOK. The technique does not require that the source, LED or laser, be capable of producing a pure frequency, which as we will see below is problematic.

The channel from source input  $x(t)$  to detector output  $y(t)$  in Figure 5-18 is simply a baseband channel, and can be used for baseband PAM transmission. OOK is a simple two-level baseband PAM signal. Multilevel baseband transmission is avoided because at the very high data rates of fiber transmission, the more complex receivers that are required are difficult to design. It has historically been easier to increase the bit rate by faster signaling using better fiber, sources, and detectors. The optical fiber channel itself has potentially such a high bandwidth that the limitation on bit rate is imposed by the electronics in the transmitter and particularly the receiver, rather than the channel. The best strategy is therefore to keep the receiver as simple as possible. Another reason for avoiding multilevel transmission is that it is difficult to control the transmitted power accurately.

In the following subsections we consider the important sources of noise in the system. Generally we will not consider ISI to be an important impairment. However, due to the much greater repeater spacings that are possible with optical amplifiers (Section 5.3.5), there has recently been a flurry of activity in the equalization of chromatic dispersion.

### 8.8.2. Quantum Limit

The received signal at the optical detector output is random due to the random arrival times of photons. This is not a flaw in the detector, but rather is a fundamental law of physics as dictated by quantum mechanics. Quantum effects are present but insignificant at microwave frequencies and below, but are important at optical frequencies. We can get some idea of the fundamental limits imposed by quantum effects by considering an ideal receiver in which all other impairments are absent, and determining the error probability. The result is a bound on the performance of OOK modulation known as the *quantum limit* [10]. (It should be noted that the quantum limit applies only to OOK modulation, although similar limits presumably can be determined for other modulation techniques.)

In OOK, a pulse of light is transmitted for a "one" bit, and no pulse is transmitted for a "zero" bit. If we were magically able to eliminate all other sources of noise, and were able to reliably detect a single photoelectron in the receiver circuitry, then we could use the following receive criterion: no pulse was transmitted ("zero" bit) if we receive zero photoelectrons, and a pulse was transmitted ("one" bit) if we receive one or more photoelectrons in the baud interval. (Recall from Section 5.3 that a photoelectron is an electron-hole pair induced in the photodetector by an incident photon.) From Section 5.3.4, if we assume no dark current ( $\lambda_0 = 0$ , see Section 5.3) and perfect quantum efficiency ( $\eta = 1$ ), the arrival rate of photoelectrons is

$$\lambda(t) = \frac{P(t)}{h\nu} \quad (8.159)$$

where  $P(t)$  is the received optical power and  $h\nu$  is the energy in one photon. The

number of photoelectrons observed during a baud interval is a Poisson distributed random variable with parameter  $\Lambda$ , where, from (3.133),  $\Lambda$  is the integral of  $\lambda(t)$  over the baud interval. Thus, when a pulse is transmitted ( $\Lambda > 0$ ) the probability of  $n$  received photoelectrons is

$$p(n) = \frac{\Lambda^n e^{-\Lambda}}{n!} . \quad (8.160)$$

Since the integral of power is energy,

$$\Lambda = \frac{E_b}{h\nu} \quad (8.161)$$

where  $E_b$  is the total received optical energy in the baud interval. Since  $h\nu$  is the energy of one photon, another interpretation of  $\Lambda$  is as the average number of photons arriving at the detector in one baud interval for a "one" bit.

We can now see why quantum effects are negligible at microwave frequencies. Since these frequencies are about five orders of magnitude smaller than optical frequencies, the energy per photon is five orders of magnitude smaller, and for a given received pulse energy the average number of photons is five orders of magnitude larger! Since the variance of the Poisson distribution in (8.160) is equal to the mean, the standard deviation is the square-root of the mean. The "width" of the distribution, as defined by the standard deviation divided by the mean, approaches zero as the mean gets large. Thus, for a very large number of received photons, the width of the Poisson distribution approaches zero and the randomness due to quantum effects becomes negligible.

Returning to the optical case, for our idealized receiver no error can be made if no pulse is transmitted, since precisely zero photons will be received. Hence, the only error that is possible results if a pulse is transmitted and no photons are observed. From (8.160), the probability of error is therefore

$$P_e = 0.5 p(0) = 0.5 \cdot e^{-\Lambda} \quad (8.162)$$

where again  $\Lambda$  is the average number of observed photons when a pulse is transmitted and the factor of 0.5 reflects the fact that no errors occur if no pulse is transmitted (we assume that input bits are equally likely). We can also write this in terms of the average number of arriving photons per bit  $M = 0.5\Lambda$ , assuming equally like "0" and "1",

$$P_e = 0.5 \cdot e^{-2M} . \quad (8.163)$$

The quantum limit relates the required average number of photoelectrons to the probability of error,

$$\Lambda = -\log_e(2P_e) . \quad (8.164)$$

or

$$M = -0.5 \log_e(2P_e) . \quad (8.165)$$

**Example 8-25.**

For an error probability of  $10^{-9}$ , we must have  $\Lambda = 20$  photoelectrons, whereas for  $10^{-6}$  only  $\Lambda = 13$  photoelectrons are required.  $\square$

It is important to note that the quantum limit is not an information-theoretic bound on the performance of the channel, but rather is a bound on the performance of an OOK detector. Other modulation schemes can theoretically achieve more than one bit per photon, analogous to the ability to achieve multiple bits/sec-Hz in spectral efficiency on radio channels. On the other hand, the quantum limit cannot be approached in a practical direct detection OOK receiver because we cannot reliably detect a signal this small in the thermal noise introduced in an electrical preamplifier. In practice we need a received power roughly 10 to 20 dB larger than this (200 to 2000 photons) [11]. These sensitivities can be improved by using an optical preamplifier rather than electrical preamplifier.

The quantum limit is useful in the same sense that the notion of channel capacity is useful — it tells us what additional performance can be achieved through heroic measures in our OOK receiver design. If all other impairments could be eliminated, then this would be the performance that could be achieved. Coherent techniques, discussed momentarily, can approach the quantum limit.

**8.8.3. Filtered Poisson and Avalanche Noise**

We will now characterize the quantum and avalanche noise in terms of its second order statistics. The terms used in this section are defined in Section 5.3. Since the PIN photodiode detector is a special case of an APD detector where the avalanche gain is unity, we will consider the latter more general case. The receiver design for optical fiber with detailed consideration of the noise analysis was pioneered in the early 1970's by S.D. Personick, then at Bell Laboratories.

A more refined model than the idealized receiver used to derive the quantum limit must account for the photodetector bias circuitry, preamplifier impulse response to a single photoelectron  $h(t)$ , and the avalanche gain  $G_m$ . The preamplifier output is therefore a random process of the form (neglecting thermal noise)

$$Y(t) = \sum_m G_m h(t - t_m) \quad (8.166)$$

where the  $t_m$  are a set of Poisson arrival times. Define the first and second moments of the avalanche gain as  $\bar{G}$  and  $\bar{G}^2$ , where from (5.41) the two are related through the excess gain factor  $F_G$ ,

$$\bar{G}^2 = F_G \bar{G}^2. \quad (8.167)$$

From Section 3.4.4, the mean and variance of the filtered Poisson process is given by

$$m_Y(t) = \bar{G} \cdot \lambda(t) * h(t) \quad \sigma_Y^2(t) = \bar{G}^2 \cdot \lambda(t) * h^2(t). \quad (8.168)$$

Even though the preamplifier output signal  $Y(t)$  is random in nature, we can model it for the purpose of second order statistics and SNR as a deterministic signal  $m_Y(t)$  with additive zero-mean noise with variance  $\sigma_Y^2(t)$ . Since  $\lambda(t)$  is proportional

to the received optical power, the preamplifier output signal is proportional to the avalanche gain  $\bar{G}$  and the received optical power. The fundamental difference from the additive Gaussian noise case considered earlier in this chapter is that the preamplifier output noise variance is also proportional to the received power. Thus, this noise has similar characteristics to crosstalk in a multiple wire-pair system, in that the noise level increases as the signal level increases. It is also similar to the quantization noise experienced in the voiceband telephone channel (Section 5.5) in this respect.

Using these results we can calculate the SNR. If we approximate the received optical power as a constant  $P$ , then the preamplifier output signal is proportional to  $\bar{G} \cdot P$  and the noise variance is proportional to  $F_G \bar{G}^2 \cdot P$ . Defining the constants of proportionality as  $\alpha$  and  $\beta$  respectively, the SNR (defined as the ratio of signal squared to noise variance) becomes

$$SNR_{PA} = \frac{\alpha^2 \bar{G}^2 \cdot P^2}{\beta F_G \bar{G}^2 \cdot P} = \frac{\alpha^2}{\beta} \cdot \frac{P}{F_G}. \quad (8.169)$$

We see that the SNR improves by one dB for each dB increase in the received power. This dependence is the same as with additive Gaussian noise, although for much different reasons! In the Gaussian noise case the noise variance stays constant and the signal squared is proportional to the signal power. In this case, the preamplifier noise has variance proportional to the optical signal power and the preamplifier signal power is proportional to the *square* of the optical signal power.

The question remains as to how to choose the avalanche gain  $\bar{G}$ . From (8.169), we can maximize the SNR by minimizing the excess gain factor  $F_G$ . From (5.42), this factor is a monotonically increasing function of the avalanche gain  $\bar{G}$ . Hence, we can maximize the SNR by letting  $\bar{G} = 1$ ; that is, using a PIN photodiode in preference to an APD as the detector. The result follows because we have not yet considered thermal noise introduced in the preamplifier. In the absence of this thermal noise, the APD detector is deleterious.

### Preamplifier Thermal Noise

As discussed in Section 5.3, another important noise source in fiber systems is thermal noise introduced in the preamplifier. This noise tends to be significant because the signal current at the output of the detector is so small, and therefore the thermal noise introduced at that point is significant relative to the signal level. This is the motivation for using an APD detector, since this is a way to boost the signal level without affecting the thermal noise level. Unfortunately this benefit comes at the price of the additional noise source due to random avalanche multiplication. Since the latter noise generally increases with avalanche gain, there is an optimum gain for any given signal and thermal noise level as we will now show.

Extending the analysis of the last section, assume that there is an additional thermal noise (within the bandwidth of interest) of  $\sigma^2$  at the preamplifier output. The result is that the SNR is now

$$SNR = \frac{\alpha^2 \bar{G}^2 P^2}{\sigma^2 + \beta F_G \bar{G}^2 P} = SNR_{PA} \cdot \frac{1}{1 + \frac{\sigma^2}{\beta F_G \bar{G}^2 P}} \quad (8.170)$$

where  $SNR_{PA}$  is given in (8.169). The second term, due to thermal noise, reduces the SNR. The thermal noise can be mitigated by either increasing the optical signal power  $P$  or by increasing the avalanche gain  $\bar{G}$ . Thus, avalanche gain is helpful in this case. However, as the avalanche gain  $\bar{G}$  is increased, the SNR must eventually start falling because the second term approaches unity and the first term,  $SNR_{PA}$ , decreases since  $F_G$  is monotonically increasing with  $\bar{G}$ .

In summary, avalanche gain is useful because it increases the signal level at the input to the preamplifier without affecting the thermal noise. However, it introduces its own excess noise in the random avalanche multiplication, and as a result there is an optimum avalanche gain above which avalanche multiplication is the dominant noise and the SNR starts to decrease again. In practical optical fiber system designs, when a PIN photodiode is used as a detector, the dominant noise source in the system is thermal noise at microwave frequencies and below, and therefore the white Gaussian noise analysis of earlier sections of this chapter is directly applicable. When an APD detector is used, then filtered Poisson noise and avalanche multiplication noise are significant impairments in addition to this thermal noise.

#### 8.8.4. Homodyne and Heterodyne Optical Reception

The number of repeaters required in a fiber optic network is inversely proportional to the bit rate of each repeater times the repeater spacing (Problem 5-10). A promising way to reduce the number of repeaters, and hence the network cost, is therefore to increase the repeater spacing. From (5.28) the way in which to increase the repeater spacing is to either reduce the fiber loss, increase the transmitted power, or reduce the received power (increase the receiver sensitivity). The fiber losses are already approaching the theoretical limit for the materials being used, about 0.2 dB per km, and the transmitted power is limited by nonlinear materials effects. Thus, we are left with the option of increasing the receiver sensitivity. For the minimum fiber loss, an improvement of 10dB in receiver sensitivity implies up to 50 additional kilometers between repeaters for fixed transmitter power. Fortunately a way to substantially increase the receiver sensitivity has been demonstrated in the laboratory; namely, *homodyne* or *heterodyne* reception.

Together these techniques are often called *coherent optical reception*, although we avoid that term here because of the possible confusion with coherent demodulation. Coherent demodulation uses a carrier at the receiver that has the same frequency and phase (approximately) as the carrier at the transmitter. In optical fiber reception the term coherent refers to the requirement for highly coherent lasers (monochromatic and relatively constant phase). In heterodyne detection the detection method may in fact be incoherent (in the sense that no attempt is made in the receiver to estimate the phase of the carrier), whereas in homodyne detection the reception must be coherent. We will see that incoherent FSK detection (Section 6.6) is among the most promising techniques for heterodyne optical fiber reception.



It has been demonstrated that homodyne receivers can approach the quantum limit in sensitivity, and heterodyne receivers give up 3 dB in sensitivity. Early work with heterodyne optical communication used lasers to transmit over large distances in space [12,13,14]. In such an application the power available to the transmitter can be quite small, so receiver sensitivity is crucial. In optical fibers, we have the luxury of being able to use regenerative repeaters, but for economic reasons we want to reduce the number of repeaters.

APD receivers are typically used for weak optical signals, but because of random fluctuations in their gain, they are noisy. In contrast, a PIN photodiode yields at most one electron-hole pair per incident photon, resulting in a weak electrical signal for small incident power and thermal noise when we amplify this signal. Homodyne and heterodyne receivers use PIN photodiodes even when the optical signal is weak. They use a local light source to supplement the incoming photons in such a way as to significantly enhance the sensitivity of the receiver. This offers the significant advantages of eliminating the APD multiplication noise and taking advantage of the higher bandwidth capabilities of the PIN diode (roughly three to four times the bandwidth of the APD).

While theoretically very interesting, heterodyne reception has not achieved commercial viability primarily because optical amplifiers offer many of the same advantages at lower cost. We will discuss homodyne detection, followed by heterodyne detection, in the following subsections.

### Homodyne Detection

Assume that the electromagnetic field at the receiver can be represented as

$$r(t) = \pm A \cos(\omega_0 t) \quad (8.171)$$

where  $A^2$  is proportional to the optical power and hence proportional to the average rate of arrival of photons. The signal is a binary antipodal PSK signal,  $+A \cos(\omega_0 t)$  representing a "one" and  $-A \cos(\omega_0 t)$  representing a "zero". The generation of such a signal requires a monochromatic light source with fixed phase, an ideal that can be approached with sufficient accuracy in practice to make the detection techniques that follow of practical interest.

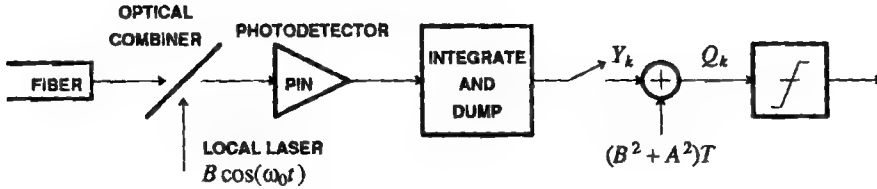
The *ideal homodyne detector* adds (optically) a local signal of exactly the same frequency and phase,

$$s(t) = B \cos(\omega_0 t) \quad (8.172)$$

getting

$$x(t) = r(t) + s(t) = (B \pm A) \cos(\omega_0 t), \quad (8.173)$$

as shown in Figure 8-14. We assume that  $B$  is much larger than  $A$ , with the result that the optical power falling on the photodetector is sufficiently large that thermal noise effects in the receiver electronics will be rendered negligible. The optical power incident to the photodetector is proportional to  $(B \pm A)^2$ , depending on whether a "one" or "zero" is transmitted, and therefore the energy and the average number of photons to arrive in symbol interval  $T$  is  $K \cdot (B \pm A)^2 T$  for some constant of



**Figure 8-14.** A coherent optical receiver works by optically adding a locally generated optical signal to the received optical signal and detecting the sum with a PIN photodiode.

proportionality  $K$ . A reasonable detection system effectively counts the number of photons arriving in each symbol interval. This can be approximated by integrating the output of a PIN photodiode and sampling at the end of the symbol interval, as shown in Figure 8-14. The integrator should be reset (*dumped*) before the next symbol interval. The expected number of photoelectrons that will be generated in the PIN photodiode will be, assuming perfect quantum efficiency,

$$\Lambda = K \cdot (B \pm A)^2 T = K \cdot (B^2 + A^2 \pm 2AB) T. \quad (8.174)$$

Since the constant  $K$  will not affect the results to follow, we will set  $K = 1$ . The receiver can subtract out the common term  $(B^2 + A^2)T$  and the result is a binary antipodal signal  $\pm 2ABT$  which can be applied to a slicer with threshold at zero.

If  $B$  is large, then the number of photons arriving at the PIN photodiode is large, so little electrical amplification is needed. Furthermore, since the desired component of the signal  $\pm 2AB$  is proportional to  $B$ , it can be made large. There is no need, therefore, to use an APD and suffer its random gain, since thermal noise can be made insignificant. This is the principal advantage of homodyne detection.

To analyze the performance of the homodyne detector we need to characterize the noise at the input to the slicer. To this end we will use the Chernoff bound to characterize the probability of error asymptotically as the average number of photoelectrons gets large, and show that the resulting probability of error is the same as the quantum limit. The following exercise gives a useful result.

### Exercise 8-3.

Use the first two terms of a Taylor series expansion,

$$\ln(1 + \epsilon) = \epsilon - 0.5\epsilon^2 \dots \quad (8.175)$$

to show that for a Poisson random variable  $X$  with large parameter  $a$  and a small  $\delta > 0$  such that  $\delta/a \ll 1$ , the Chernov bound of Problem 3-19 becomes

$$1 - F_X(a + \delta) \leq e^{-\delta^2/2a} \quad (8.176)$$

$$F_X(a - \delta) \leq e^{-\delta^2/2a} \quad (8.177)$$

□

Neglecting thermal noise, the random variable  $Y_k$  at the output of the integrate-and-dump filter is Poisson distributed with parameter  $a = \Lambda$  given by (8.174). The probability of error assuming the signal is  $A \cos(\omega_c t)$  is  $F_X(\Lambda - \delta)$  where  $\delta = 2ABT$ , and the Chernoff bound for this is given by (8.177). In this bound we can approximate  $\Lambda$  by  $B^2T$  for large  $B$ , in which case the bound becomes

$$P_e \leq e^{-2A^2T} \quad (8.178)$$

Using (8.176) to bound the probability of error assuming signal  $-A \cos(\omega_c t)$  is transmitted, we get the same answer as (8.178), and hence (8.178) becomes the upper bound on the probability of error regardless of the *a priori* probability of each signal. We can relate this probability of error to the average number of received photons  $M$ , since  $M = A^2T$  regardless of which signal is transmitted, and hence the Chernoff bound becomes

$$P_e \leq e^{-2M} \quad (8.179)$$

the same as the quantum limit in (8.163) if we disregard the insignificant factor of 0.5. As shown in Section 6.5, for small probability of error this multiplicative factor does not result in a significant difference in the signal power required to achieve a given probability of error, so we see that ideal homodyne detection permits us to closely approximate the quantum limit as the local oscillator amplitude  $B$  gets large. Stated another way, an ideal homodyne detector for 2-PSK performs as well as an ideal photon counter with OOK when the average receive power is the same, but shows more promise of being practical. Note that if we constrain the *peak* power instead of the average power, then the ideal homodyne detector actually performs 3dB *better* than the quantum limit (see Problem 8-21).

## Heterodyne Detection

Heterodyne detection is similar to homodyne except that the local laser has a frequency different than the carrier, thus achieving a frequency translation to an *intermediate frequency* (IF) rather than directly to baseband. Although heterodyne techniques have been common in radio applications, it was not until 1955 that the first beat signal from the mixing of two light sources on a photocathode was reported.

If the output of the local laser is written

$$s(t) = B \cos(\omega_1 t) \quad (8.180)$$

then the sum of the incoming and local optical signals for a single symbol, taken without loss of generality as occurring over the interval  $0 \leq t \leq T$ , is

$$x(t) = \pm A \cos(\omega_0 t) + B \cos(\omega_1 t) \quad (8.181)$$

### Exercise 8-4.

Show that (8.181) can be written in terms of the envelope and phase about a carrier at frequency  $\omega_1$  as

$$x(t) = E(t) \cos(\omega_1 t + \beta(t)) \quad (8.182)$$

where the envelope is

$$E^2(t) = B^2 + A^2 \pm 2AB \cos(\omega_{IF}t) , \quad (8.183)$$

$\omega_{IF} = \omega_0 - \omega_1$  is the intermediate frequency (IF), and  $\beta(t)$  is a time-varying phase.  $\square$

The photon arrivals form a Poisson process with arrival rate  $\lambda(t)$  proportional to the square of the envelope, which is the instantaneous power,

$$\lambda(t) = E^2(t) . \quad (8.184)$$

The output of the photodiode and preamplifier electronics will be a shot noise process. If we let  $h(t)$  be the impulse response of the photodiode bias circuitry and preamplifier, then from (3.141) (with  $\beta = 1$  and  $\lambda(t)$  large) the preamplifier output approaches a Gaussian process with mean value

$$s(t) = \lambda(t) * h(t) \quad (8.185)$$

and variance

$$\sigma^2(t) = \lambda(t) * h^2(t) . \quad (8.186)$$

We expect the bandwidth of the preamplifier to be large relative to the IF, and hence  $s(t) \approx \lambda(t)$  assuming unity gain in the passband. Ignoring the d.c. term, which will be subtracted prior to the slicer as in the homodyne case,

$$s(t) \approx \pm 2AB \cos(\omega_{IF}t) . \quad (8.187)$$

Similarly, since  $B$  is large, for purposes of the noise variance we can consider  $\lambda(t) \approx B^2$ , and the noise variance is therefore independent of time,

$$\sigma^2 = B^2 \int_{-\infty}^{\infty} h^2(t) dt . \quad (8.188)$$

#### Example 8-26.

If the preamplifier has a flat gain with bandwidth  $W$  radians/sec, we get  $\sigma^2 = B^2 W / \pi$ . Since the noise variance is proportional to bandwidth, we can consider this noise to be white Gaussian noise passed through a bandlimiting filter. If the bandwidth is large relative to IF, we can consider the additive Gaussian noise to be white with power spectral density  $B^2$ .  $\square$

On the basis of this example, we will assume the additive noise  $N(t)$  to be white and Gaussian with power spectral density  $B^2$ , and write the reception in the form

$$Y(t) = \pm 2AB \cos(\omega_{IF}t) + N(t) . \quad (8.189)$$

Put into this approximate form, the detection problem now is identical to that of 2-PSK as discussed earlier in this Chapter. A correlation or matched filter receiver (justified intuitively in Section 6.6 and shown to be optimal in a maximum likelihood sense in Chapter 8) computes the correlation

$$Q = \int_0^T Y(t) \cos(\omega_{IF}t) dt \quad (8.190)$$

and decides that a one was sent if  $Q$  is greater than zero and a zero was sent otherwise.

**Exercise 8-5.**

Show that the signal to noise ratio in  $Q$ , the input to the slicer, is given by

$$SNR = \frac{(ABT)^2}{B^2 T/2} = 2A^2 T. \quad (8.191)$$

Assume that  $\omega_{IF}$  is either large or satisfies  $\omega_{IF}T = K2\pi$  where  $K$  is an integer.  $\square$

The probability of error for this antipodal symbol set is

$$P_e = Q(\sqrt{SNR}) = Q(\sqrt{2A^2 T}) \leq e^{-A^2 T} = e^{-M} \quad (8.192)$$

where  $M = A^2 T$  is again the average number of photons per bit arriving over the fiber. The inequality again follows from the Chernoff bound (3.43). The exponent is a factor of two smaller than for homodyne detection implying that heterodyne detection requires a 3 dB higher signal power than the quantum limit or ideal homodyne detection.

In spite of this penalty for heterodyne detection, it is very attractive from a practical perspective. Heterodyne detection does not require the local laser to be precise in either frequency or phase, since uncertainties in the IF can be compensated by carrier recovery applied to the IF signal. In addition, heterodyne detection allows incoherent demodulation of FSK or MSK signals, obviating the need for carrier recovery at IF (at the expense of a small additional penalty in SNR).

Perhaps the most intriguing possibility for heterodyne detection is *optical frequency-division multiplexing*, in which many closely spaced carriers are used to transmit independent data streams. Frequency-division multiplexing can be used with direct detection also (this is called *wavelength-division multiplexing (WDM)*) but the much larger bandwidth of the intensity-modulated data stream makes it much less bandwidth efficient. The bandwidth efficiency of WDM is on the order of  $10^{-6}$  b/s/Hz, as compared to about  $10^{-1}$  for heterodyne detection. While bandwidth is not by any means a scarce resource in the optical fiber medium, if the maximum repeater spacing and bit rate are to be obtained we must limit the bandwidth of the optical signal to regions of low attenuation and small chromatic dispersion, making optical FDM a very attractive approach.

**Laser Phase Noise**

Phase or frequency noise in lasers phenomenon seriously complicates homodyne and heterodyne fiber detection [15,16,17]. Laser phase noise is caused by randomly occurring spontaneous emission events. Each event causes a spontaneous jump (of random magnitude and sign) in the phase of the electromagnetic output. The phase executes a random walk away from the value it would have in the absence of spontaneous emission. As the time between events becomes very small, the phase due to the events can be approximated as the integral of a white Gaussian noise process

$$\Theta(t) = 2\pi \int_0^t N(t) dt \quad (8.193)$$

where  $N(t)$  has power spectrum  $N_0$ . The power spectrum of  $N(t)$  is a property of the

laser.

Laser phase noise is observable as a broadening of the spectrum of the output of the laser. The 3dB width of the spectrum of the laser is called its *linewidth*. The lasers most likely to be used in optical fiber systems, semiconductor injection lasers, can be made with linewidths in the range of 5 to 50 Mhz.

#### Example 8-27.

The linewidth required for heterodyne detection with MSK incoherent demodulation is approximately  $10^{-3}f_d$ , where  $f_d$  is the bit rate. For homodyne detection with PSK modulation the required linewidth is about  $10^{-4}f_d$ . This more stringent requirement stems from the required coherent demodulation of PSK. A laser with linewidth of 10 Mhz will therefore support a bit rate of about 10 GHz (heterodyne MSK detection) or 1 GHz (homodyne PSK detection). □

Considerable research effort is devoted to designing appropriate lasers with much narrower linewidths. Until these lasers become available, the most promising coherent fiber signaling scheme is heterodyne FSK or MSK with incoherent demodulation.

## 8.9. FURTHER READING

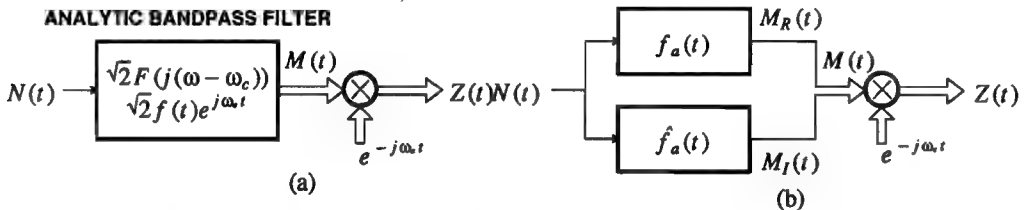
The geometric approach to estimating the error probability of modulation systems was originally inspired by the text by Wozencraft and Jacobs [4]. The approach to comparing modulation systems used here, based on the normalized signal-to-noise ratio, was heavily influenced by [9].

Spread-spectrum is covered in some detail in the digital communications textbooks by Proakis [18], Cooper and McGillem [19], and Ziemer and Peterson [20]. Our coverage has relied heavily on two excellent tutorial articles [6,21].

The design of fiber optic receivers, with a detailed analysis, is covered more thoroughly in the books written by Personick [22,10], Barnoski [23], and Gagliardi and Karp [24]. A useful overview of optical fiber technology is given by Henry [25]. A study of practical limits for direct detection is given by Pierce [26]. For coherent fiber optics, see the survey articles by Salz [27] and Barry and Lee [28]. A general description is also given by Kimura [29], accompanied by several excellent papers in the special issue of the *IEEE Journal of Lightwave Technology* in April 1987, jointly prepared with the *IEEE Journal on Selected Areas in Communications*. For example, one paper in the issue is on optical continuous-phase FSK by Iwashita and Matsumoto [30]. A thorough analysis of the bit error rate of various coherent optical receivers is given by Okoshi *et. al.* [31]. Kazovsky gives an excellent comparison of optical heterodyne vs. homodyne receivers [32] as well as an analysis of the impact of laser phase noise on heterodyne systems [33]. Other theoretical analyses of various types of coherent receivers of note are [34,31,35].

## PROBLEMS

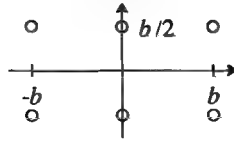
- 8-1. In this problem we derive the statistics of the noise  $Z(t)$  at the output of a PAM receiver receive filter in a different way from Section 8.3. The configuration we will use is shown in Figure 6-19a. The front end of this receiver is reproduced in Figure 8-15, where the input is the noise  $N(t)$ , the complex-valued noise at the output of the analytic bandpass filter is denoted by  $M(t)$ , and after demodulation by  $Z(t)$ .
- Assuming that  $f(t)$  is bandlimited to  $\omega_c$ , what can you say about the relationship of the real and imaginary parts of  $\sqrt{2}f(t)e^{j\omega_c t}$ ?
  - Explicitly write the real and imaginary parts of  $\sqrt{2}f(t)e^{j\omega_c t}$  in terms of  $f(t)$ .
  - What are the variances of the real and imaginary parts of  $M(t)$ , as well as their cross-correlation, in terms of  $f(t)$  and  $N_0$ ?
  - Use the results of (c) to show that the real and imaginary parts of  $M(t)$  are independent and have the same variance.
  - Show that  $Z(t)$  has the same first-order statistics as  $M(t)$ .
- 8-2. Assume that the real-valued receive filter  $f(t)$  in Figure 8-15 is an ideal lowpass filter with bandwidth  $W$  radians/sec and that the symbol rate obeys  $\pi/T = W$ . Show that the noise samples at the slicer input  $Z(kT)$  are white in this case.
- 8-3. Compare  $Q(d/2\sigma)$  and  $Q^2(d/2\sigma)$  for values  $d = 2$  and  $\sigma = 0.5$ . Do it again for  $\sigma = 0.25$ . Is the approximation in (8.62) valid for these values of  $\sigma$ ? You may use Figure 3-1 to approximate  $Q(\cdot)$ .
- 8-4. Consider the 4-PSK constellation in Figure 7-5. Assume that  $\sigma = 0.25$  is the standard deviation of the independent real and imaginary parts of additive Gaussian noise. Assume  $b = 1$  and the transmitted symbol is  $-1$ . Find the probability that the received sample is closer to  $j$  than to  $-1$  and compare it to the probability that the received sample is closer to  $+1$  than to  $-1$ . You may use Figure 3-1 to estimate the probabilities.
- 8-5. Show that the probability of error for the 16-QAM constellation of Figure 7-5 can be written
- $$\text{Pr}[\text{error}] = 3Q(d/2\sigma) - 2.25Q^2(d/2\sigma). \quad (8.194)$$
- 8-6. In this problem we put together the results of Chapter 6 and 8 to analyze a passband system. Assume a benign channel,  $B(j\omega) = 1$ , with additive Gaussian noise with power spectrum  $S_N(j\omega) = N_0$ . The transmit filter produces a 100% excess-bandwidth raised-cosine pulse. The transmit power cannot be greater than unity. The receive filter has ideal lowpass baseband equivalent  $f(t)$  that permits the 100% excess-bandwidth pulse to get through undistorted. The



**Figure 8-15.** Configuration for calculating the noise at receive filter output. a) A complex-valued filter realization, and b) a detail showing explicitly the real and imaginary parts of  $M(t)$ .

constellation is 16-QAM. Find the probability of error as a function of  $N_0$  and  $T$ .

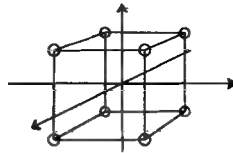
- 8-7. Consider the constellation in the following figure:



Assume that

- The inner two symbols each have probability  $1/4$ .
- The outer four symbols each have probability  $1/8$ .
- The noise in each dimension is independent and Gaussian with variance  $\sigma^2$ .

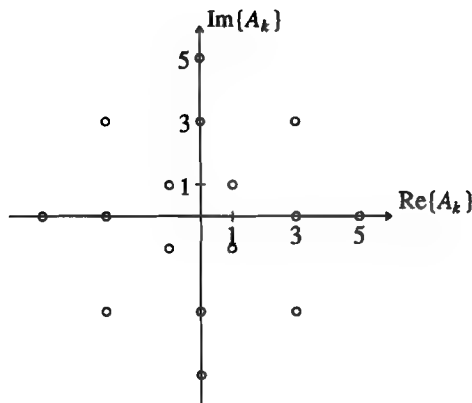
- (a) Design a coder for this constellation that achieves these probabilities if the input bits are equally likely to be zero or one.
  - (b) Find the exact probability of error as a function of  $b$ .
  - (c) Find the signal power as a function of  $b$ .
  - (d) Give the probability of error as function of the SNR. Use an approximation from Figure 3-1 to find the probability of error when  $SNR = 10dB$ .
  - (e) Give approximations for the probability of error. Compute the approximate probabilities of error when  $SNR = 10dB$ .
- 8-8. Suppose that more than two dimensions are available for our alphabet. Consider an alphabet where the symbols are vertices of an  $M$  dimensional hypercube, shown for  $M = 3$  in the following figure:



Assume that all the symbols are equally likely and the noise in each dimension is independent with variance  $\sigma^2$ .

- (a) What are the decision regions?
  - (b) What is the probability of error as a function of  $M$  and the minimum distance  $d$  between points in the signal constellation.
- 8-9. Assume a constellation that is  $M$  dimensional, consisting of  $M$  equally likely symbols at right angles each with magnitude  $a$ . Find a bound on the probability of error using the union bound.
- 8-10.
- (a) Find the union bound on the probability of error for the 16-QAM constellation in Figure 7-5b. Assume  $A_k = c + jc$  is actually transmitted.
  - (b) The CCITT V.29 standard for full-duplex transmission at 9600 b/s over voiceband channels uses the constellation shown in Figure 8-16. Find the union bound on the probability of error. Assume  $A_k = 1 + j$  is transmitted.
  - (c) Explain why the exact analysis technique of Example 8-6 would be difficult to apply for the V.29 constellation.
  - (d) Find  $c$  in Figure 7-5 so that the two constellations have the same power. Use the union bounds of parts (a) and (b) to compare their performance.





**Figure 8-16.** The constellation for the CCITT V.29 standard for transmission at 9600 b/s over voiceband channels.

- 8-11. Show that if the translates of the chip waveform  $h_c(t)$  are mutually orthogonal, then  $N = 2BT$  pulses of the form of (8.113) can be made mutually orthogonal by choice of the spreading sequences. Specify the required properties of the spreading sequences.
- 8-12. Consider the spreading sequence  $\{x_m, 0 \leq m \leq N-1\}$  in (8.113) to be the impulse response of a causal, discrete-time FIR filter. Condition (8.116) suggests that we would like this filter to be an allpass filter. Use the results of Section 2.5.3 to show that the only FIR filters that are allpass have impulse response  $\delta_{k-L}$ , for some integer  $L$ . Thus, (8.116) can be exactly satisfied only for the trivial choice of spreading sequence in Example 8-13.
- 8-13. Consider a spread-spectrum system operating in  $N = 2BT$  dimensional signal space, where the isolated pulse signal is chosen randomly. Let a set of orthonormal basis functions for this space be  $\phi_i(t)$ ,  $1 \leq i \leq N$ . The one-dimensional binary antipodal transmitted signal is chosen to be

$$\pm S(t) = \pm \sum_{i=1}^N S_i \phi_i(t) \quad (8.195)$$

and a jammer generates a similar signal

$$J(t) = \sum_{i=1}^N J_i \phi_i(t) \quad (8.196)$$

where the  $S_i$  and  $J_i$  are mutually independent zero-mean random variables. The signal components are chosen to have variance  $E[S_i^2] = \sigma_k^2/N$  where  $\sigma_k^2$  is the average energy per bit, and the jammer chooses the  $J_i$  variances to satisfy the constraint

$$\sum_{i=1}^N E[J_i^2] = E_J \quad (8.197)$$

where  $E_J$  is the constrained jamming signal energy per symbol interval. The receiver is told the  $S_i$ , and applies a matched filter to the received signal.

- Determine the signal and noise random variables at the output of the matched filter.
- Define the SNR at the matched filter output as the ratio of the mean-signal squared to the noise variance. Show that this SNR has a processing gain of  $N$  independently of how the jammer distributes its energy among the signal components. (This result is due to the fact that the signal is truly random and unknown to the jammer.)

- 8-14.** Verify that the capacity of the passband channel in Figure 8-6b is given by (8.137). Use a signal space argument with complex rather than real signals, noise, and vectors.
- 8-15.** One way to view digital communication is as a way in which to exchange channel bandwidth for improved noise immunity. For example, in analog modulation systems, FM modulation is thought of primarily as a way to achieve higher post-demodulation SNR in exchange for increased bandwidth. In this problem we quantify this tradeoff for digital communication systems, using SNR for a given channel capacity as a measure of noise immunity. Given two white Gaussian noise channels as in Figure 8-6, with bandwidths  $B_1$  and  $B_2$ , where  $B_2 > B_1$ , suppose they have the same channel capacity. Find a relation for the SNRs (in dB) required for the two channels as a function of the bandwidth expansion factor  $B_2/B_1$ . Interpret this relation. You may assume large SNR.
- 8-16.** It is common to use binary PSK in spread spectrum modulation. Find the SNR gap to capacity for 2-PSK spread spectrum as a function of  $BT$ . What is the increase in the SNR gap to capacity asymptotically as  $BT \rightarrow \infty$ , expressed in dB?
- 8-17.** Consider the input to the slicer in a direct-detection optical fiber receiver with a PIN detector, which consists of a Poisson random variable with mean-value  $\Lambda_0$  or  $\Lambda_1$  for the two possible signals plus independent additive Gaussian noise with variance  $\sigma^2$ . Find the Chernoff bound on the probability of error, assuming the two signals are equally likely.
- 8-18.** In this problem we will explore the conditions under which the direct-detection OOK optical-fiber receiver performance is limited by thermal noise. Assume a PIN detector with 100% quantum efficiency, a bit rate of  $10^8$  bits/sec, and a wavelength of  $1.5 \mu\text{m}$ . Assume the front end of the receiver consists of a current source (the photodetector) in series with a 10K ohm resistor, and the voltage across the resistor is integrated for each baud interval and applied to a slicer (integrate-and-dump receiver). The 10K ohm resistor has an internal thermal noise source.
- What is the incident average optical power required to achieve a  $10^{-9}$  error probability at the quantum limit?
  - Use the results of Problem 5-12 to find the variance of the thermal noise component of the slicer input. Also find the size of the average signal component at the slicer input for a transmitted one bit.
  - At what incident average optical power is the signal to thermal noise ratio (for a one bit) at the slicer input equal to 20 dB?
  - At the incident optical power of (c), how many photons per one bit are incident on the detector?
  - What are the relative sizes of the variance of the shot noise and thermal noise at the slicer input for the incident optical power of (c)?
- 8-19.** For the same conditions as Problem 8-18, adjust the incident optical power so that the signal to thermal noise ratio at the slicer input is only 10 dB.
- Using an APD detector with ionization ratio  $k = .03$ , find the APD gain that maximizes the total SNR, including both thermal and shot noise components, at the slicer input (this will require a numerical solution, with the aid of a calculator or computer).
  - For this optimum APD gain, how much of a gain in SNR is attributable to the APD relative to a PIN detector?
- 8-20.** In the ideal homodyne optical detector, the larger  $B$  gets in (8.172), the more dynamic range is required in the electrical circuits prior to the subtraction in Figure 8-14. The purpose of this problem is to show that the dynamic range requirement is modest. Let the local laser produce 1000 times as many photons as are arriving from the fiber, so  $B^2 = 1000A^2$ . Assuming there is no noise, find the ratio (in dB) of the power of the desired signal  $\pm 2ABT$  at the sampler and the power of the common term  $(B^2 + A^2)T$  that is subtracted out.

- 8-21. Show that an ideal homodyne 2-PSK detector performs at least 3 dB better than an ideal OOK photon counting receiver (the quantum limit) if the *peak* received power is the same in both systems.
- 8-22. Repeat the derivation of the Chernoff bound of (8.179) using the following technique. Use the fact that a Poisson random variable with large parameter  $a$  approaches a Gaussian random variable, and then approximate the probability of error using the Chernoff bound for a Gaussian random variable in (3.43).

## REFERENCES

1. D. B. Williams and D. H. Johnson, "On Resolving 2M-1 Narrow-Band Signals with an M Sensor Uniform Linear Array," *IEEE Trans. on Signal Processing*, p. 707 (March 1992).
2. N. R. Goodman, "Statistical Analysis based on a Certain Multivariate Complex Gaussian Distribution (An Introduction)," *The Annals of Mathematical Statistics* 34(1) pp. 152-177 (March 1963).
3. S.W. Golomb, *Digital Communications with Space Applications*, Prentice Hall, N.J. (1964).
4. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York (1965).
5. R. A. Scholtz, "The Origins of Spread-Spectrum Communications," *IEEE Trans. Communications* COM-30(5) p. 822 (May 1982).
6. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread-Spectrum Communications — A Tutorial," *IEEE Trans. Communications* COM-30(5) p. 855 (May 1982).
7. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Illinois (1963).
8. C. E. Shannon, "Communication in the Presence of Noise," *Proc. IRE* 37 pp. 10-21 (Jan. 1949).
9. G. D. Forney, Jr and M. V. Eyuboglu, "Combined Equalization and Coding Using Precoding," *IEEE Communications Magazine*, (Dec. 1991).
10. S. D. Personick, *Fiber Optics Technology and Applications*, Plenum Press, New York (1985).
11. J. C. Campbell, A. G. Dentai, W. S. Holden, and B. L. Kasper, "High Performance Avalanche Photodiode with Separate Absorption, Grading, and Multiplication Regions," *Elect. Lett.* 19 pp. 818-819 (Sep. 29, 1983).
12. M. Ross, *Laser Receivers*, John C. Wiley and Sons (1966).
13. O. E. DeLangue, "Optical Heterodyne Detection," *IEEE Spectrum*, pp. 77-85 (Oct. 1968).
14. W. K. Pratt, *Laser Communication Systems*, John C. Wiley and Sons (1969).
15. C. H. Henry, "Theory of the Linewidth of Semiconductor Lasers," *IEEE J. Quant. Elec* QE-18 pp. 259-264 (Feb. 1982).
16. M. W. Fleming and A. Mooradian, "Fundamental Line Broadening of Single-Mode GaAlAs Diode Lasers," *Appl. Phys. Lett.* 38 pp. 511-513 (April 1, 1981).
17. C. Harder, K. Vahala, and A. Yariv, "Measurement of the Linewidths Enhancement Factor  $\alpha$  of Semiconductor Lasers," *Appl. Phys. Lett.* 42 pp. 328-330 (Feb. 15, 1983).
18. J. G. Proakis, *Digital Communications, Second Edition*, McGraw-Hill Book Co., New York (1989).
19. G. R. Cooper and C. D. McGillem, *Modern Communications and Spread Spectrum*, McGraw-Hill Book Co., New York (1986).

20. R. E. Ziemer and R. L. Peterson, *Digital Communications and Spread Spectrum Systems*, Macmillan, New York (1985).
21. C. E. Cook and H. S. Marsh, "An Introduction to Spread Spectrum," *IEEE Communications Magazine*, p.8, (March 1983).
22. S. D. Personick, *Optical Fiber Transmission Systems*, Plenum Press, New York (1981).
23. M. K. Barnoski, *Fundamentals of Optical Fiber Communications*, Academic Press, New York (1976).
24. R. Gagliardi and S. Karp, *Optical Communications*, Wiley-Interscience, New York (1976).
25. P. S. Henry, "Introduction to Lightwave Transmission," *IEEE Communications* 23(5)(May 1985).
26. J. Pierce, "Optical Channels: Practical Limits with Photon Counting," *IEEE Trans. on Communications*, (Dec. 1978).
27. J. Salz, "Modulations and Detection for Coherent Lightwave Communications," *IEEE Communications Magazine* 24(6)(June 1986).
28. J. R. Barry and E. A. Lee., "Performance of Coherent Optical Receivers," *Proceedings of the IEEE* 78(8)(Aug. 1990).
29. T. Kimura, "Coherent Optical Fiber Transmission," *IEEE/OSA Journal of Lightwave Technology* LT-5(4)(April 1987).
30. K. Iwashita and T. Matsumoto, "Modulation and Detection Characteristics of Optical Continuous Phase FSK Transmission System," *IEEE/OSA Journal of Lightwave Technology* LT-5(4)(April 1987).
31. T. Okoshi, K. Emura, K. Kikuchi, and R. Th. Kersten, "Computation of Bit-Error Rate of Various Heterodyne and Coherent-Type Optical Communication Schemes," *J. Optical Communications* 2 pp. 89-96 (1981).
32. L. G. Kazovsky, "Optical Heterodyning Versus Optical Homodyning: A Comparison," *J. Opt. Commun.* 6(1) pp. 18-24 (1985).
33. L. G. Kazovsky, "Impact of Laser Phase Noise on Optical Heterodyne Communication Systems," *J. Opt. Commun.* 7(2) pp. 66-78 (1986).
34. Y. Yamamoto and T. Kimura, "Coherent Optical Fiber Transmission Systems," *IEEE J. Quantum Electronics* QE-17(6) pp. 919-935 (June 1981).
35. T. Okoshi, "Heterodyne and Coherent Optical Fiber Communications: Recent Progress," *IEEE Trans. on Micr. Th. and Tech.* MTT-30 pp. 1138-1148 (Aug. 1982).

# 9

---

## DETECTION

---

We saw in Chapter 8 that one of the fundamental problems in digital communications is the corruption of the transmitted signal by noise. Using common sense, practical receivers can be designed that are reasonably robust in the presence of noise. Nevertheless, the question arises: Are the "common sense" receivers designed in Chapter 6 and 7 optimal? In this chapter we develop a theory of optimal detection for both discrete-time and continuous-time channels. With this theory, we will verify that the receiver structures given in Chapters 6 and 7 are optimal under certain circumstances and certain criteria of optimality. Chapters 13 and 14 will also use the theory we develop here to decode error-correction and trellis codes. In fact, we will uncover an underlying commonality between the problem of detection of data symbols on channels with intersymbol interference and trellis decoding.

The general approach to deriving optimal receivers is to model the relationship between the transmitted and received signals by a joint probability distribution. Based on the noisy observation (the received signal plus noise), we wish to *estimate* or *detect* the input signal. We use the term *estimation* when the transmitted signal is a continuous-valued random variable, as is often the case in an analog communication system, and the term *detection* when the transmitted signal is discrete-valued (even if the received signal is continuous-valued). The primary distinction is that in detection we can often recover the signal exactly with high probability, a restatement of the regeneration principle of Chapter 1. In estimation, by contrast, we must be satisfied with a recovered signal that may be more accurate than the observation but will not be exact. In this chapter we study only detection, although very similar techniques can

be applied to the estimation problems of analog communications. In fact, we will encounter *parameter estimation* problems in Chapter 11 when we communicate over channels with parameters initially unknown.

In order to address the detection problem, we need a statistical model for the received signal. Before the data symbols arrive at the detector, they are processed by a transmitter, pass through a channel, and are further processed by the front end of the receiver. Some of this processing is deterministic, such as any filtering functions, and some is random, such as additive noise on the channel. In this chapter we call the deterministic portion *signal generation* and a random component *noise generation*. The model is shown in Figure 9-1. The input  $X_k$  is a discrete-time and discrete-valued random process. It is not only discrete-valued but has a *finite* number of possible values, each of which is a function of the source bits.

**Example 9-1.**

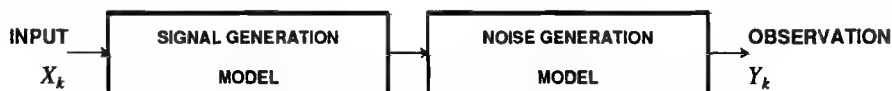
Suppose  $X_k$  is a data symbol sequence. Then the signal generator could be a discrete-time equivalent transmit filter and channel transfer function, and the noise generator could be *additive Gaussian noise*, independent of  $X_k$ . □

**Example 9-2.**

Suppose  $X_k$  is a bit sequence. The signal generator could be a coder that produces another bit sequence, and the noise generator could be modeled as a *binary symmetric channel* (BSC), which randomly inverts some of the bits. □

The receiver uses the *observation*  $Y_k$  to make a *decision* about  $X_k$ . The observation can be either discrete or continuous-valued. Since each  $X_k$  has a finite number of possible values, the detector must make a decision from among a finite number of alternatives.

Two related detection methods are covered: *maximum likelihood (ML)* and *maximum a-posteriori probability (MAP)*. MAP detection, also called *Bayesian* detection, is optimal in the sense that it minimizes the probability of error. While probability of error is undoubtedly the most appropriate criterion to minimize in most digital communications systems, ML detection is almost always used in practice instead of MAP detection. ML detection is a special case of MAP detection for the simplifying assumption that all the possible inputs are equally likely. It is often reasonable to assume equally likely signals, and in any case the performance of the simpler ML detector is usually so close to that of the MAP detector that there is little incentive to



**Figure 9-1.** A signal  $X_k$  to be transmitted is processed deterministically (signal generation) and stochastically (noise generation).

implement a costlier MAP detector.

We begin with simple signal generation models and progress to more realistic (and more complicated) models. We address the two basic noise generators of Example 9-1 and Example 9-2 — additive Gaussian noise and the BSC. We begin with the detection of a single real-valued data symbol, where the input is a real-valued data symbol and the signal generator is trivial. Then we progress to the detection of vector-valued inputs, which applies to the case of complex-valued signal constellations among others. At this point we will have theoretically justified the slicer used so liberally in Chapter 6, at least for the case where there is no intersymbol interference (ISI). The next step is to derive the optimal detector in additive Gaussian noise, for both the discrete-time and continuous-time cases. This is followed by relaxing the known-signal assumption by allowing the carrier phase to be unknown (random). Up to this point the optimal detectors have been defined for the detection of a single data symbol, and the next extension is to ISI, where it is shown that the minimum-distance receiver design of Section 7.4 is optimal in additive Gaussian noise. A low-complexity algorithm for carrying out the minimization with all data-symbol sequences, the Viterbi algorithm, is then derived. The Viterbi algorithm has many other applications in digital communication, including the detection of convolutional and trellis codes (Chapters 13 and 14). Finally, the detection of a shot-noise signal with known intensity, characteristic of fiber optic systems, is considered.

## 9.1. DETECTION OF A SINGLE REAL-VALUED SYMBOL

In this section, we consider the simplest case, where the input is a single random variable  $X$  (a single data symbol  $A$ ) rather than a random process, and the signal generator passes this symbol directly through to the noise generator without modification. The data symbol has as a sample space the alphabet  $\Omega_A$ , as discussed in Chapter 6. Noise generators that arise in practice for this case result in either a discrete-valued observation  $Y$  or a continuous-valued observation. We give examples of both cases in the following subsections, at the same time illustrating the ML and MAP detectors.

### 9.1.1. Discrete-Valued Observations

Some noise generators result in discrete-valued observations  $Y$ . In order to design a detector, we must know the discrete distribution of  $Y$  conditioned on knowledge of the data symbol,  $p_{Y|A}(y|\hat{a})$ , as this completely specifies the noise generator. The *maximum likelihood* (ML) detector chooses  $\hat{a} \in \Omega_A$  to maximize the *likelihood*  $p_{Y|A}(y|\hat{a})$ , where  $y$  is the observed outcome of  $Y$ .

#### Example 9-3.

Suppose that we have additive discrete noise  $N$ , so that  $Y = A + N$ . Assume  $A$  and  $N$  are independent and take on values zero and one according to the flip of two fair coins. There are three possible observations,  $y = 0, 1$ , or  $2$ . The likelihoods for the observation  $y = 0$  are

$$p_{Y|A}(0|0) = 0.5 \quad (9.1)$$

$$p_{Y|A}(0|1) = 0 \quad (9.2)$$

so if the observation is  $y = 0$ , the ML detector selects  $\hat{d} = 0$ . If the observation is  $y = 1$ , then the likelihoods are equal

$$p_{Y|A}(1|0) = p_{Y|A}(1|1) = 0.5, \quad (9.3)$$

so the ML detector selects either zero or one (we could pick at random, for example). If the observation is  $y = 2$ , the ML detector selects  $\hat{d} = 1$ .  $\square$

The advantage of ML detection is that the likelihood  $p_{Y|A}(y|\hat{d})$  is easily computed for each possible  $\hat{d}$  knowing only the statistics of the noise generator and not the statistics of the data symbol.

The *maximum a-posteriori probability* (MAP) detector maximizes the *posterior probability*  $p_{A|Y}(\hat{d}|y)$ . This should be more intuitively appealing than maximizing the likelihood. In fact, we will see that the MAP receiver minimizes the probability of error, and hence is optimal for any application that prefers correct decisions over incorrect. We can write the posterior probability in terms of the likelihood using Bayes' rule (3.32)

$$p_{A|Y}(\hat{d}|y) = \frac{p_{Y|A}(y|\hat{d})p_A(\hat{d})}{p_Y(y)}. \quad (9.4)$$

A MAP detector therefore needs to know the probabilities  $p_A(\cdot)$  of the symbols. These probabilities are called the *prior probabilities*, the *a priori probabilities*, or more simply, just the *prior*. Since Bayes' rule is used to compute the posterior probability, MAP detection is also called *Bayesian detection*. Since  $p_Y(y)$  does not depend on the detector decision  $\hat{d}$ , maximizing the posterior probability is the same as maximizing  $p_{Y|A}(y|\hat{d})p_A(\hat{d})$ . If  $p_A(\hat{d})$  is constant for all  $\hat{d} \in \Omega_A$ , then maximizing the posterior probability is the same as maximizing the likelihood  $p_{Y|A}(y|\hat{d})$ , and MAP reduces to ML.

#### Example 9-4.

In Example 9-3,  $p_A(\hat{d}) = 0.5$ , for each possible  $\hat{d}$ , so MAP and ML are equivalent.  $\square$

#### Example 9-5.

If we knew *a-priori* in Example 9-3 that one of the coins was biased (unfair), say

$$p_A(0) = 0.75 \quad (9.5)$$

$$p_A(1) = 0.25, \quad (9.6)$$

then the MAP detector would not generally give the same result as the ML detector. It is easily shown that if the observation is  $y = 0$  or  $y = 2$ , the ML or MAP detections are the same and are correct with probability one. If the observation is  $y = 1$ , however, the detectors are not the same. In this case, the likelihoods are



$$p_{Y|A}(1|\hat{a}) = \begin{cases} 0.5; & \text{for } \hat{a} = 1 \\ 0.5; & \text{for } \hat{a} = 0 \end{cases} \quad (9.7)$$

so the ML detector can arbitrarily select a decision. However, when  $y = 1$ , the posterior probabilities are

$$p_{A|Y}(\hat{a}|1) = \begin{cases} 0.25; & \text{for } \hat{a} = 1 \\ 0.75; & \text{for } \hat{a} = 0 \end{cases} \quad (9.8)$$

so the MAP detector always gives  $\hat{a} = 0$  when the observation is  $y = 1$ .  $\square$

Whenever we are detecting a signal based on a noisy observation there is of course the possibility of making an error or mistake. The *probability of error*

$$P_e = \Pr[\hat{a} \neq a] \quad (9.9)$$

is a good measure of the quality of our detector design.

#### Example 9-6.

In Example 9-3, when the coin flip is fair, the MAP and ML detectors are identical, so their probability of error is identical. We can compute that probability of error. The observation  $Y = A + N$  takes on values in  $\{0, 1, 2\}$  with probabilities  $\{0.25, 0.5, 0.25\}$ , respectively. If the observation is  $y = 0$ , then both detectors give the result  $\hat{a} = 0$ , and no error occurs. Similarly, if  $y = 2$ , both detectors give  $\hat{a} = 1$ , and no error occurs. If the observation is  $y = 1$ , however, both detectors are unable to choose a unique maximum. If the detectors select  $\hat{a}$  to randomly equal zero or one, then they will be wrong half the time. Since the  $p_Y(1) = 0.5$ , the total probability of error is  $P_e = 0.25$ .  $\square$

In the case of equal prior probabilities for  $A$ , ML and MAP detectors give identical results, and their probability of error is identical. This is no longer true if the priors are different. In this case, a MAP detector will always have a lower probability of error than a ML detector.

#### Example 9-7.

Assume that the data symbol  $A$  in Example 9-3 is generated by an unfair coin, just as in Example 9-5. We again assume a noisy observation  $Y = A + N$ , where  $N$  is a fair coin. If the observation is either  $y = 0$  or  $y = 2$ , then both detectors are correct, and no error is made. If  $y = 1$  and the ML detector arbitrarily selects a decision, it will make errors with probability 0.5 each time it observes  $y = 1$ . This occurs with probability  $p_Y(1) = 0.5$ , so the total probability of error is 0.25. The MAP detector, however, will do much better, because when  $y = 1$  it always selects  $\hat{a} = 0$ . This is incorrect for only  $1/4$  of such observations, and  $p_Y(1) = 0.5$ , so the probability of error has been reduced to 0.125.  $\square$

In fact, the MAP detector minimizes the probability of error. To show this, note the probability of a correct decision is

$$\Pr[\text{correct decision}] = \sum_{y \in \Omega_Y} \Pr[\text{correct decision} | Y = y] p_Y(y). \quad (9.10)$$

Since  $p_Y(y) \geq 0$ , the probability of a correct decision is maximized if the decision for each observation  $y$  maximizes  $\Pr[\text{correct decision} | Y = y]$ . This is what the MAP receiver does since the probability of a correct decision is

$$\Pr[\text{correct decision} | Y = y] = p_{A|Y}(\hat{a} | y), \quad (9.11)$$

and this is precisely the posterior probability maximized by the MAP detector.

In summary, ML and MAP are different detection techniques, but yield the same result when the prior probabilities are equal. If the prior probabilities are different, the MAP detector will yield a lower (and indeed the minimum) probability of error. If we have no information about the prior probabilities, we usually assume they are equal, and MAP reduces to ML.

The most common noise generation model in digital communications that results in a discrete-valued observation is the *binary symmetric channel (BSC)*. This channel applies when the input data symbol is binary, and the conditional probabilities are shown in Figure 9-2. The noisy observation differs from the binary input symbol with probability  $p$ , which we call the error probability of the BSC. The ML and MAP detectors for a BSC are derived in Problem 9-1.

### 9.1.2. Continuous-Valued Observations

A common noise generation model corrupts the input data symbol  $A$  by *additive continuous-valued* noise  $N$ . The observation  $Y = A + N$  is then a continuous-valued random variable. The MAP detector selects  $\hat{a}$  to maximize  $p_{A|Y}(\hat{a} | y)$  while the ML detector selects  $\hat{a}$  to maximize  $f_{Y|A}(y | \hat{a})$ . Given an observation  $Y = y$ , to find the MAP detector we use the mixed form of Bayes' rule (3.31) to write

$$p_{A|Y}(\hat{a} | y) = \frac{f_{Y|A}(y | \hat{a})p_A(\hat{a})}{f_Y(y)}. \quad (9.12)$$

The denominator is independent of the decision, so we need only maximize the numerator. This is illustrated by example.

#### Example 9-8.

Suppose that  $A$  takes on value  $\pm 1$  according to the flip of an unfair coin, so that  $p_A(+1) = 0.75$  and  $p_A(-1) = 0.25$ . Suppose further that we observe  $Y = A + N$  where  $N$  is a continuous-valued random variable with the p.d.f. given in Figure 9-3a. We can derive the MAP detector and its probability of error. We need to select  $\hat{a}$  to maximize  $f_{Y|A}(y | \hat{a})p_A(\hat{a})$ . In Figure 9-3b, this quantity is shown for the two possible values of  $\hat{a}$  as a function of the observation. From this figure it is immediately evident that the MAP

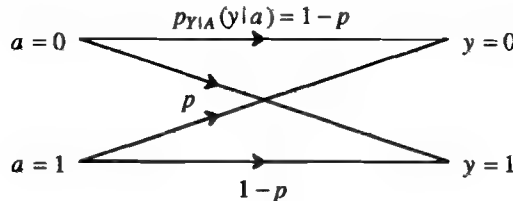
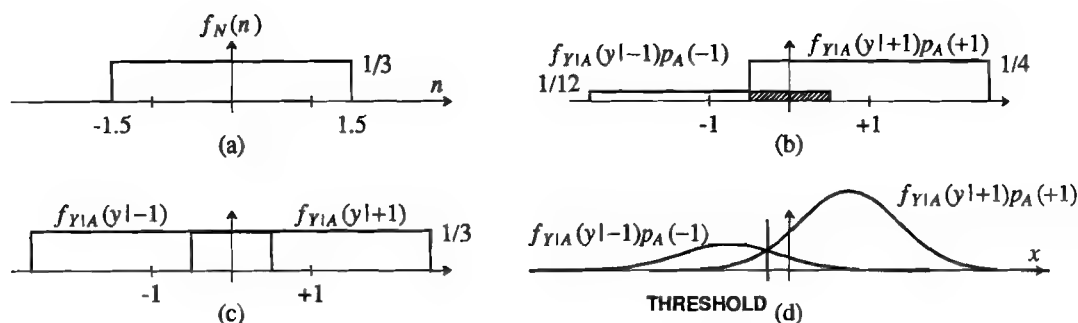


Figure 9-2. The transition probabilities of the binary symmetric channel.



**Figure 9-3.** (a) The p.d.f. of uniformly distributed noise. (b) For  $\hat{d} = \pm 1$ , the MAP criterion  $f_{Y|A}(y|\hat{d})p_A(\hat{d})$  is plotted. Note that the MAP detector will prefer  $\hat{d} = +1$  for any observation  $y > -0.5$ . (c) The likelihoods as a function of the observation  $y$  are plotted for  $\hat{d} = \pm 1$ . Note that the ML detector has no preference when the observation is  $-0.5 < y < +0.5$ . (d) The MAP criterion is plotted assuming additive Gaussian noise.

detector will set  $\hat{d} = +1$  if  $y > -0.5$ , otherwise  $\hat{d} = -1$ . An error never occurs if  $A = +1$ . An error occurs one third of the time that  $-1$  is transmitted, because one third of the time the observation will be greater than  $-0.5$ . Hence, the probability of error is  $p_A(-1)/3 = 1/12$ . This is the area of the shaded region Figure 9-3b.  $\square$

### Example 9-9.

We now find the ML detector and its probability of error for the same scenario as in the previous example. In Figure 9-3c we plot the likelihoods for the two possible decisions as a function of the observations. The likelihoods are equal in the region  $-0.5 < y < 0.5$ , so the ML detector can make its decision arbitrarily. A legal ML detector sets its threshold at  $-0.5$ , and has performance identical to that of the MAP detector. But another legal ML detector sets its threshold at zero (halfway between the two possibilities); this detector will make an error  $1/6$  of the time for each possible transmission, so the probability of error is  $1/6$ .  $\square$

The most common distribution for additive noise in digital communications is Gaussian, rather than uniform as in previous examples. The principle of the detectors is the same.

### Example 9-10.

In Figure 9-3d we show the functions  $f_{Y|A}(y|\pm 1)p_A(\pm 1)$  as functions of the observations assuming additive Gaussian noise. For the MAP detector, the threshold is selected where these curves cross. For the ML detector, the threshold is selected at zero.  $\square$

In the next section we will consider the additive Gaussian noise case for the more general situation where the signal and noise are vector-valued. This will model many situations that we encounter in digital communications.

## 9.2. DETECTION OF A SIGNAL VECTOR

Many of the communication channels we describe in this book can be modeled as noise corrupting a vector-valued signal. Although typical channels accept only scalar-valued signals, a convenient vector communication model can often be obtained using the technique shown in Figure 9-4. A vector of transmitted symbols is converted to a sequence of scalars for transmission over an additive noise channel, and then reconverted to a vector at the channel output. In effect we have taken a finite sequence of samples and modeled them as a vector.

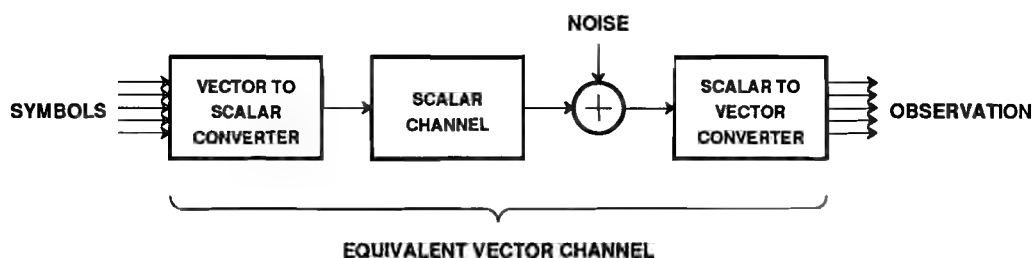
### Example 9-11.

In Section 8.2 we formulated a vector-valued received signal consisting of a set of known signals and additive Gaussian noise. Further, we showed in Section 8.3 that this formulation applied directly to the minimum-distance receiver design considered in Chapter 7, if the appropriate decision variables were calculated based on an orthonormal expansion of the subspace of known signals. The PAM slicer design considered in Section 8.3 was a special one-dimensional case.  $\square$

### Example 9-12.

A signal generator that results in a binary signal vector appropriate as input to a BSC (Figure 9-2) is a *binary block code*, to be considered in Chapters 12 and 13. In this case,  $\mathbf{S}$  is a vector of values "0" or "1". If  $\mathbf{S}$  is transmitted over a BSC, some of these bits may be inverted. We will show that the ML detector selects the codeword "closest" to the received bits in the sense that the fewest number of bits are different. Binary block codes are often used for error detection and error correction (Chapter 13).  $\square$

A general model for the situation is as follows. The signal generator accepts an input  $X$  and maps it into a vector signal  $\mathbf{S}$  with dimension  $M$ .



**Figure 9-4.** A scalar channel plus some signal processing can sometimes be modeled as a vector channel.

**Example 9-13.**

The signal generator might take a set of  $M$  consecutive data symbols as the signal vector,  $\mathbf{S} = [A_1, \dots, A_M]$ .  $\square$

The observation is a vector  $\mathbf{Y}$  with the same dimension as the signal. The noise generator is specified by the conditional distribution of the observation given the signal,  $f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}|\mathbf{s})$ . The detector decides which signal vector  $\hat{\mathbf{s}}$  from among all the possible signal vectors was actually transmitted based on the observation. A common characteristic of the noise generator is *independent noise components*, by which we mean precisely that

$$f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}|\mathbf{s}) = \prod_{k=1}^M f_{Y_k|S_k}(y_k|s_k) , \quad (9.13)$$

or in words, given knowledge of the signal vector, each component of the noise generation is independent of the others.

In the following subsections we will consider first the ML detector (for which the signal generator does not need to be statistically characterized) and then the MAP detector.

**9.2.1. ML Detection**

The ML detector chooses the signal vector  $\hat{\mathbf{s}}$  from among all the possibilities in order to maximize the conditional probability  $f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}|\hat{\mathbf{s}})$ , a probability given directly by the noise generation model. It is simple in that the statistics of the signal  $\mathbf{S}$  need not be taken into account. Two important examples are the additive Gaussian noise generator and the BSC noise model.

**Example 9-14.**

Consider the additive Gaussian noise problem formulated in (8.20), where the complex-valued noise vector  $\mathbf{Z}$  is assumed to be circularly symmetric with uncorrelated (and hence independent) components with variance  $2\sigma^2$ . The received signal  $\mathbf{Y}$  is therefore a complex-valued Gaussian vector with mean equal to  $\mathbf{s}$ , and hence has the probability density function

$$f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}|\hat{\mathbf{s}}) = f_{\mathbf{Z}|\mathbf{S}}(\mathbf{y} - \hat{\mathbf{s}}|\hat{\mathbf{s}}) = f_{\mathbf{Z}}(\mathbf{y} - \hat{\mathbf{s}}) . \quad (9.14)$$

Hence the ML detector selects  $\hat{\mathbf{s}}$  to maximize  $f_{\mathbf{Z}}(\mathbf{y} - \hat{\mathbf{s}})$ . Since the components of  $\mathbf{Z}$  are assumed Gaussian and independent, from (3.49) we obtain

$$f_{\mathbf{Z}}(\mathbf{z}) = \prod_{k=1}^K f_{Z_k}(\mathbf{z}_k) . \quad (9.15)$$

Since, for a complex Gaussian random variable with independent real and imaginary components,

$$f_{Z_k}(\mathbf{z}_k) = \frac{1}{2\pi\sigma^2} e^{-|\mathbf{z}_k|^2/2\sigma^2} , \quad (9.16)$$

it follows that

$$f_Z(\mathbf{z}) = \frac{1}{(2\pi)^K \sigma^{2K}} e^{-\|\mathbf{z}\|^2/2\sigma^2}. \quad (9.17)$$

Since the exponential is a monotonic function of its exponent, maximizing  $f_Z(\mathbf{y} - \hat{\mathbf{s}})$  is thus equivalent to minimizing  $\|\mathbf{z}\| = \|\mathbf{y} - \hat{\mathbf{s}}\|^2$ . Thus, the ML detector is equivalent to a minimum-distance detector, as considered in Section 8.2.  $\square$

### Example 9-15.

For the binary symmetric channel (BSC) of Figure 9-2 with independent noise components, the components of the signal vector are binary, as are the components of the observation. The conditional probability for one channel use is

$$p_{Y_k|S_k}(\mathbf{y}|\mathbf{s}) = \begin{cases} p, & \mathbf{y} \neq \mathbf{s} \\ 1-p, & \mathbf{y} = \mathbf{s} \end{cases} \quad (9.18)$$

Define the metric  $d_H(\hat{\mathbf{s}}, \mathbf{y})$  as the number of components in which the two binary vectors  $\hat{\mathbf{s}}$  and  $\mathbf{y}$  disagree. This metric  $d_H(\cdot, \cdot)$  is so fundamental to coding theory that it is given the special name *Hamming distance* or *Hamming metric* after R. Hamming of Bell Laboratories, who did pioneering work on algebraic coding in the 1950's. If the vectors have dimension  $M$ , then the joint conditional distribution is

$$p_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}|\hat{\mathbf{s}}) = p^{d_H(\hat{\mathbf{s}}, \mathbf{y})} (1-p)^{M-d_H(\hat{\mathbf{s}}, \mathbf{y})} = (1-p)^M \cdot \left[ \frac{p}{1-p} \right]^{d_H(\hat{\mathbf{s}}, \mathbf{y})}. \quad (9.19)$$

When  $p < 1/2$  as is usual, the ML detector chooses  $\hat{\mathbf{s}}$  to minimize  $d_H(\hat{\mathbf{s}}, \mathbf{y})$ . In other words, it chooses the signal vector closest to the observation vector in Hamming distance.  $\square$

The two important noise generators with independent additive noise components, the additive Gaussian noise and the BSC, result in a similar ML detector: choose that signal vector which is the closest to the observation vector. The only difference between the two cases is the manner in which we measure "distance"; in the first case we use Euclidean distance and in the other we use Hamming distance. In both cases, the sample space  $\Omega_Y$  can be divided into *decision regions*. The  $i^{\text{th}}$  decision region is the set of all  $\mathbf{y}$  closer to  $\mathbf{s}_i \in \Omega_S$  (in Euclidean or Hamming distance) than to any other  $\mathbf{s}_j$ ,  $j \neq i$ . In other words, it is the set of all observations that will lead to the decision  $\hat{\mathbf{s}} = \mathbf{s}_i$ .

### Example 9-16.

Let the signal  $\mathbf{s}$  be a two-dimensional vector from the set

$$\left\{ \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\}. \quad (9.20)$$

The signal space and decision regions are shown in Figure 7-5a.  $\square$

The ML detector in both cases is simple and intuitive, and has the feature that the noise variance  $\sigma^2$  or error probability  $p$  need not be known by the detector. This feature is desirable, since in many digital communications systems the noise variance or error probability may be dependent on factors such as the distance of transmission, and therefore difficult to know in advance.

### 9.2.2. MAP Detector

The MAP detector is considerably more complicated and requires knowledge of the noise variance (Gaussian channel) or error probability (BSC). Assume that the prior probabilities  $p_S(\hat{s})$  are known for each possible signal  $\hat{s}$ . Given the observation  $y$ , the MAP detector selects  $\hat{s}$  to maximize the posterior probability

$$p_{S|Y}(\hat{s}|y) = \frac{f_{Y|S}(y|\hat{s})p_S(\hat{s})}{f_Y(y)}. \quad (9.21)$$

The denominator is independent of  $\hat{s}$ , so the MAP detector equivalently selects  $\hat{s}$  to maximize the numerator,  $f_{Y|S}(y|\hat{s})p_S(\hat{s})$ . If the prior probabilities  $p_S(\hat{s})$  are equal (the signals are equally likely), then the MAP detector reduces to the ML detector, regardless of the channel parameters.

#### Example 9-17.

For the additive Gaussian noise channel of Example 9-14, the MAP detector maximizes

$$f_Z(y - \hat{s})p_S(\hat{s}) = \frac{1}{(2\pi)^M \sigma^{2M}} \exp \left[ -\frac{1}{2\sigma^2} \|y - \hat{s}\|^2 \right] p_S(\hat{s}). \quad (9.22)$$

Taking the natural logarithm, a monotonic function, we see that this is equivalent to *minimizing*

$$\|y - \hat{s}\|^2 - 2\sigma^2 \ln[p_S(\hat{s})]. \quad (9.23)$$

The sample space for  $Y$  can again be divided into decision regions, but the boundaries of the regions are not determined strictly on the basis of Euclidean distance if the prior probabilities are not equal, and unfortunately depend on the noise variance  $\sigma^2$  and the prior probabilities (see Problem 9-3 and Problem 9-4).  $\square$

A similar result occurs in the case of the BSC of Example 9-15; namely, the detector does not merely minimize the Hamming distance but rather takes into account the error probability  $p$  and the prior probabilities if they are not equal (see Problem 9-2).

In practice, ML detectors are used in digital communication systems, both because of the increased complexity of the MAP detectors, and because equal prior probabilities is the normal assumption. It is also often the case that the prior probabilities or the noise variance are not known with sufficient accuracy to implement a MAP detector. When the noise is Gaussian and the probability of error is low, the improvement in the probability of error for using a MAP detector in preference to a ML detector is not usually significant. When the MAP detector is used, the decision boundaries can be set for the worst case noise variance  $\sigma^2$ , thereby giving the optimal performance in the worst case. In more critical applications the noise variance can be estimated.

### 9.2.3. Probability of Error for BSC ML Detector

The probability of error was considered in Section 8.2 for the Gaussian noise case and minimum-distance receiver design. In this section, we have shown that the ML detector uses a minimum-distance criterion. Thus, the error probability

determined in Section 8.2 applies directly to the ML detector for the Gaussian detection problem formulated there. In this section, we extend this result to the BSC channel. Basically all that differs is the definition of distance and the definition of the  $Q(\cdot)$  function.

### Two-Signal Case

Consider two binary vectors  $s_i$  and  $s_j$  that differ in  $d$  components (the Hamming distance is  $d$ ). Suppose  $s_i$  is transmitted through a BSC noise generator. The conditional probability of the received bits, given the transmitted bits, has the form given by (9.19). From Example 9-15, the ML detector will choose  $s_j$  instead of  $s_i$  if the received bits  $y$  are closer in Hamming distance to  $s_j$  than to  $s_i$ . Any error that occurs in a component for which the bits in  $s_i$  and  $s_j$  are the same will not affect ML detection, since it will impact the Hamming distances to both  $s_j$  and  $s_i$  equally. Pessimistically assume that a detection error is always made if the received bits  $y$  are equidistant from  $s_i$  and  $s_j$ . A detection error then occurs if more than  $t$  errors occur in the  $d$  bits that differ, where

$$t = \begin{cases} (d-1)/2; & \text{if } d \text{ is odd} \\ (d/2) - 1; & \text{if } d \text{ is even} \end{cases} \quad (9.24)$$

The number of errors in  $d$  components is binomial, so the probability that the ML detector prefers  $s_j$  over  $s_i$  is

$$Q(d, p) = \sum_{i=t+1}^d \binom{d}{i} p^i (1-p)^{d-i} \quad (9.25)$$

We choose the notation  $Q(d, p)$  for this sum in order to emphasize the similarity to the  $Q(\cdot)$  function used in the Gaussian case. By convention, whenever a function  $Q$  has two arguments, we mean (9.25), and whenever it has one argument, we mean (3.38).

#### Example 9-18.

Suppose that  $s_i = [000000]$  and  $s_j = [110111]$ . The Hamming distance is  $d = 5$ . If  $s_i$  is transmitted over a BSC with error probability  $p$ , the ML detector will mistake it for  $s_j$  if three or more of the received bits in the five differing positions are changed by the channel. The probability of this occurring is

$$\begin{aligned} Q(5, p) &= \binom{5}{3} p^3 (1-p)^2 + \binom{5}{4} p^4 (1-p) + \binom{5}{5} p^5 \\ &= 10p^3(1-p)^2 + 5p^4(1-p) + p^5. \end{aligned} \quad (9.26)$$

□



### Three or More Signals

In Section 8.2, we derived an upper bound (the union bound) and a lower bound on the error probability. That same bound applies to the ML detector for the BSC channel with a redefinition of distance and the definition of the  $Q(\cdot)$  function. Just as these bounds are tight for small  $\sigma$  in the Gaussian case, they become tight for small  $p$  in the BSC case.

$Q(d, p)$  is monotonic in  $d$  for the BSC, just as  $Q(d/2\sigma)$  is monotonic in  $d$  for the Gaussian channel. Consequently, exactly the same bounds apply, where  $d_{\min}$  is Euclidean distance for the Gaussian channel, and  $d_{\min}$  is Hamming distance for the BSC. The conclusion is that for small  $p$ , the error probability is

$$P_e \approx C \cdot Q(d_{\min}, p) \quad (9.27)$$

where  $d_{\min}$  is the minimum Hamming distance among all pairs of transmitted signal vectors.

## 9.3. KNOWN SIGNALS IN GAUSSIAN NOISE

In this section we consider the problem of designing the ML detector for one of  $L$  signals in additive Gaussian noise, first for the discrete-time case and then for the continuous-time case. The discrete-time case for white noise follows in straightforward fashion from the results of Section 8.2. Our main concern will be with extending this result to nonwhite Gaussian noise, and subsequently to continuous time. For generality, we will treat the case of complex-valued noise and received signals. Subsequently we will apply the results directly to the baseband and passband signals of specific interest.

### 9.3.1. Discrete-Time Received Signal

A discrete-time received signal is of the form

$$Y_k = s_{m,k} + Z_k, \quad 0 \leq k < \infty, \quad (9.28)$$

where  $\{s_{l,k}, 1 \leq l \leq L\}$  is a set of  $L$  known signals,  $s_{m,k}$  is one of those signals, and  $Z_k$  is additive zero-mean Gaussian noise. All quantities in (9.28) are assumed to be complex-valued.

#### White Noise Case

Assume the noise is white with variance  $2\sigma^2$  and circularly symmetric (Section 8.1), and hence is also stationary. This is similar to the vector signal case considered in Section 8.2 and 9.2, except that the number of components in the vector is countably infinite. We can handle this by using the previous structure for the  $N$ -dimensional vector and allowing  $N \rightarrow \infty$ . The ML detector then calculates the Euclidean distance between the received signal and each known signal. Hence, it calculates

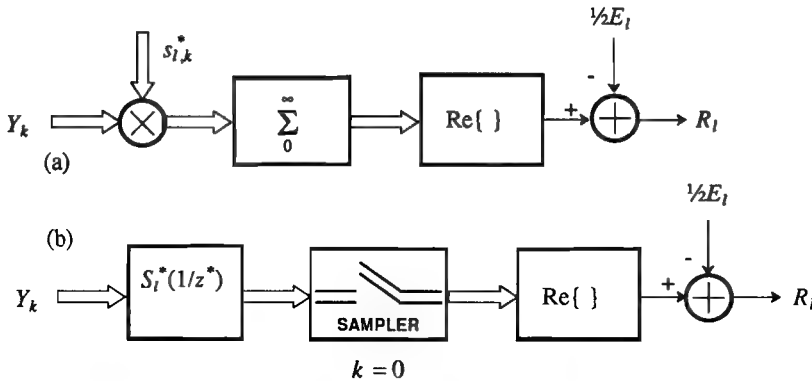
$$D_l = \sum_{k=0}^{\infty} |Y_k - s_{l,k}|^2 = \sum_{k=0}^{\infty} |Y_k|^2 + \sum_{k=0}^{\infty} |s_{l,k}|^2 - 2 \operatorname{Re} \left\{ \sum_{k=0}^{\infty} Y_k s_{l,k}^* \right\}, \quad (9.29)$$

for  $1 \leq l \leq L$ , and then chooses the  $l$  for which  $D_l$  is *minimum*. The first term is not a function of  $l$ , so it cannot affect the result and can be ignored. Thus, the ML criterion is equivalent to *maximizing*

$$R_l = \operatorname{Re} \left\{ \sum_{k=0}^{\infty} Y_k s_{l,k}^* \right\} - \frac{1}{2} \cdot E_l, \quad E_l = \sum_{k=0}^{\infty} |s_{l,k}|^2, \quad (9.30)$$

where  $E_l$  is the energy of the  $l$ -th signal. This detector crosscorrelates the received signal against the conjugate of each of the known signals  $\{s_{l,k}\}$ ,  $1 \leq l \leq L$ , and then takes the real part of the result. This is repeated for each  $l$ , and the decision is the  $l$  for which it is maximum. This interpretation of the receiver is shown in Figure 9-5a. This is a discrete-time version of the crosscorrelation receiver structure that was seen earlier in Section 6.6, 6.8, and 7.2, except that we have now generalized to arbitrary modulation formats. In addition, Figure 9-5 amounts to a generalization to signal sets for which the energy  $E_l$  is not constant.

An equivalent way to generate  $R_l$  is to apply the received signal to a filter with impulse response  $s_{l,-k}^*$ , which has transfer function  $S_l^*(1/z^*)$ , and sample the output at  $k = 0$ . This interpretation of the receiver is shown in Figure 9-5b. The filter  $S_l^*(1/z^*)$  is a discrete-time version of the matched filter first encountered in Section 6.6. As in the continuous-time case in Chapter 7, the discrete-time matched filter is anticausal. If it happens to be FIR, then it can be realized as a causal filter plus a delay. If it is IIR, then it can only be approximated in practice.



**Figure 9-5.** Two interpretations for the ML detector for a discrete-time known signal in white Gaussian noise. (a) A crosscorrelator, and (b) a discrete-time matched filter.

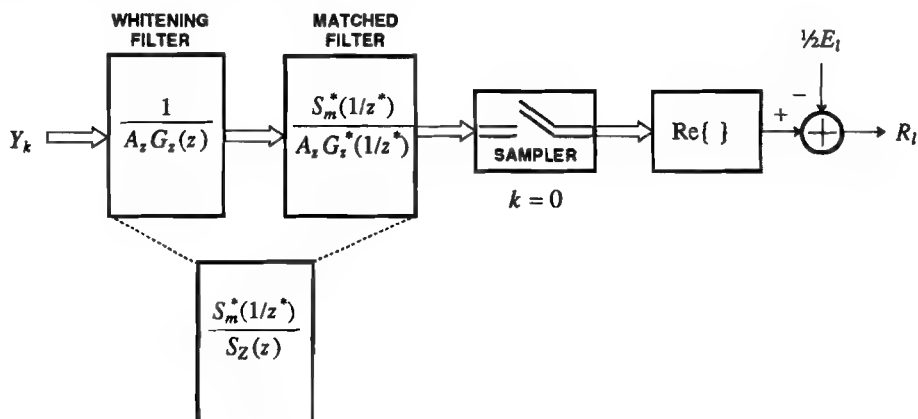
### Nonwhite Noise Case

Consider now general (possibly nonwhite) wide-sense stationary circularly symmetric Gaussian noise with power spectrum  $S_Z(z)$ . The receiver of Figure 9-5 is the ML detector for white noise only, so it cannot be directly applied to the nonwhite noise. A useful trick is to first apply the received signal to a *noise whitening filter*.

The power spectrum  $S_Z(z)$  must be non-negative real on the unit circle. In Section 2.5.2 we derived the spectral factorization of a rational  $S_Z(z)$  that is real-valued and non-negative on the unit circle,

$$S_Z(z) = A_z^2 G_z(z) G_z^*(1/z^*) \quad (9.31)$$

where  $G_z(z)$  is a loosely minimum-phase transfer function. This factorization generalizes to non-rational  $S_Z(z)$  as well. Assume that  $S_Z(z)$  has no zeros on the unit circle. In that case,  $1/G_z(z)$  is causal, stable, and strictly minimum-phase. As shown in Section 3.2.3 and illustrated again in Figure 9-6, the received signal can be passed through the strictly minimum-phase whitening filter  $1/A_z G_z(z)$ , where the noise at the output is unit-variance Gaussian. The whitening filter also affects the signal, and the Z-transform of the new signal is  $S_m(z)/A_z G_z(z)$ , where  $S_m(z)$  is the Z transform of  $\{s_{m,k}, 0 \leq k < \infty\}$ . The matched-filter receiver of Figure 9-5 can be applied to this whitened received signal, taking into account the new signal spectrum. Since linear filtering preserves the circular symmetry of Gaussian noise, the white noise is circularly symmetric, and thus the samples of the noise are mutually independent. As also shown in Figure 9-6, the whitening and matched filters can be combined into a single filter, equivalent to that in Figure 9-5 except that the transfer function is normalized by the noise spectrum  $S_Z(z)$ . This normalization is appropriate for a matched filter in nonwhite noise, and the result is an ML detector.



**Figure 9-6.** The minimum-phase whitening filter for the noise spectrum  $S_Z(z)$  whitens the noise, allowing the matched filter detector to be applied to the new signal. The whitening filter is a reversible operation, and hence will not adversely affect the error probability of the detector. Combining the two filters, we get an equivalent matched filter for nonwhite noise.

### 9.3.2. Continuous-time Reception

Suppose we have a continuous-time received signal,

$$Y(t) = s_m(t) + Z(t), \quad 0 \leq t \leq T, \quad (9.32)$$

where  $\{s_l(t), 1 \leq l \leq L, 0 \leq t \leq T\}$  is a set of known signals. The noise  $Z(t)$  is assumed to be a zero-mean circularly symmetric stationary Gaussian process with power spectrum  $S_Z(j\omega)$  and autocorrelation function  $R_Z(\tau)$ . For reasons of mathematical tractability, we address a finite (but arbitrarily large) time interval  $0 \leq t \leq T$ .

To develop the ML detector as we have in the discrete-time case, some new techniques have to be introduced. Our approach is to turn this into a problem equivalent to the discrete-time case by using a signal-space expansion of the random process  $Z(t)$ ,  $0 \leq t \leq T$ , in terms of a countable set of functions,

$$Z(t) = \sum_{i=1}^{\infty} Z_i \phi_i(t), \quad 0 \leq t \leq T \quad (9.33)$$

where the functions are orthonormal in signal space,

$$\int_0^T \phi_i(t) \phi_j^*(t) dt = \delta_{i-j}. \quad (9.34)$$

Equality in (9.33) is in the sense of mean-square convergence of the series on the right to the process on the left. This expansion is more useful if the coefficients are themselves uncorrelated,

$$E[Z_i Z_j^*] = \sigma_i^2 \delta_{i-j}. \quad (9.35)$$

Under quite general conditions, a set of orthonormal functions  $\{\phi_i(t)\}$  can be found such that (9.35) is satisfied (even for the case where  $Z(t)$  is not wide-sense stationary as we assume here). The resulting expansion is known as the *Karhunen-Loeve expansion*.

First, taking the inner product of both sides of (9.33) with  $\phi_j(t)$ ,

$$Z_j = \int_0^T Z(t) \phi_j^*(t) dt. \quad (9.36)$$

Since  $Z_j$  is a linear function of a Gaussian process, it is a Gaussian random variable, and further it is circularly symmetric since  $Z(t)$  is assumed circularly symmetric. This circular symmetry together with (9.35) implies that the  $Z_j$  are statistically independent. In Appendix 9-A, it is shown that a necessary and sufficient condition on  $\{\phi_j(t)\}$  for (9.35) to be satisfied is

$$\int_0^T R_Z(t - \tau) \phi_j(\tau) d\tau = \sigma_j^2 \phi_j(t), \quad 1 \leq j < \infty, \quad 0 \leq t \leq T. \quad (9.37)$$

Although the left side of (9.37) looks like a convolution, the equality is valid only for the finite time interval  $0 \leq t \leq T$ , so it is in fact not a convolution. This is an *integral equation*, and, in analogy to similar matrix equations,  $\phi_j(t)$  is called an *eigenfunction*

of  $R_Z(t)$  with corresponding *eigenvalue*  $\sigma_j^2$ .

The question arises as to whether there exists a set of complete orthonormal eigenfunctions and corresponding non-zero eigenvalues  $\{\phi_j(t), \sigma_j, 1 \leq j < \infty\}$  satisfying (9.37). This is considered in some detail by Van Trees [1], where it is confirmed that they do exist under rather general conditions. For our purposes, it suffices that the power spectrum  $S_Z(j\omega)$  be non-zero for all  $\omega$ . Fortunately, in the following we don't actually have to find the eigenfunctions satisfying (9.37); it suffices to know that they exist.

Now returning to the original detection problem of (9.32), the approach is to expand the received signal in the same set of orthonormal functions that arise out of the Karhunen-Loeve expansion of  $Z(t)$ ,

$$Y(t) = \sum_{i=1}^{\infty} Y_i \phi_i(t), \quad Y_i = s_{m,i} + Z_i, \quad 0 \leq t \leq T. \quad (9.38)$$

The coefficients  $Z_i$  are uncorrelated, circularly symmetric (and hence independent) Gaussian random variables. Their variances are not necessarily equal,  $E[|Z_i|^2] = \sigma_i^2$ , and the  $s_{m,i}$  are the coefficients of  $s_m(t)$  with respect to the orthonormal basis functions  $\phi_i(t)$ ,

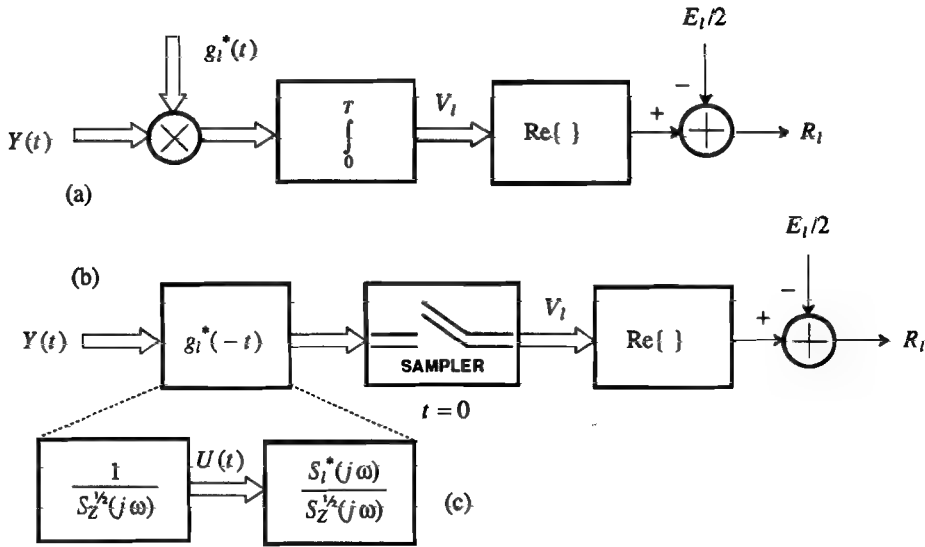
$$s_{m,i} = \int_0^T s_m(t) \phi_i^*(t) dt. \quad (9.39)$$

The Karhunen-Loeve expansion turns the continuous-time received signal into an equivalent discrete-time received signal, at least in a mathematical if not literal sense (since the  $i$  in  $Y_i$  is not time, but an index over the signal-space basis). The continuous-time received signal  $Y(t)$  is represented on the finite time interval  $0 \leq t \leq T$  by the countable set of random variables  $\{Y_i, 1 \leq i < \infty\}$ . We can apply the earlier discrete-time results to this equivalent representation, with the slight complication that the noise samples do not all have equal variance. Assuming that the eigenvalues are all non-zero, this problem is easily circumvented by normalizing the samples by dividing both sides by the known standard deviation,

$$\frac{Y_i}{\sigma_i} = \frac{s_{m,i}}{\sigma_i} + \frac{Z_i}{\sigma_i}, \quad 1 \leq i < \infty. \quad (9.40)$$

The  $Z_i/\sigma_i$  are all unit-variance Gaussian random variables. The normalization of (9.40) can be considered a form of whitening, similar to the whitening filter applied in Figure 9-6.

The normalized representation of (9.40) satisfies all the conditions assumed for the discrete-time case, namely a set of known discrete-time signals with additive white noise variables. We can therefore apply the earlier detector to the normalized received signal  $Y_i/\sigma_i$ , where the known signal component is  $s_{l,i}/\sigma_i$ . Thus, the ML detector minimizes



**Figure 9-7.** The ML detector for a continuous-time known signal in additive Gaussian noise. (a) The correlation receiver. (b) The matched filter receiver. (c) The matched filter in the limit as  $T \rightarrow \infty$ .

$$D_l = \sum_{i=1}^{\infty} \left| \frac{Y_i}{\sigma_i} - \frac{s_{l,i}}{\sigma_i} \right|^2 = \sum_{i=1}^{\infty} \frac{|Y_i - s_{l,i}|^2}{\sigma_i^2}, \quad (9.41)$$

over all possible signals  $1 \leq l \leq L$ . As in the discrete-time case, the first term  $\sum_{i=1}^{\infty} |Y_i|^2$  will be independent of  $l$ , so the ML detector equivalently maximizes the decision variable

$$R_l = \operatorname{Re} \left\{ \sum_{i=1}^{\infty} \frac{Y_i s_{l,i}^*}{\sigma_i^2} \right\} - \frac{1}{2} E_l, \quad E_l = \sum_{i=1}^{\infty} \frac{|s_{l,i}|^2}{\sigma_i^2}. \quad (9.42)$$

In Appendix 9-A, this result is related to the original continuous-time signals. In particular, defining a function  $g_l(t)$  that satisfies the integral equation

$$\int_0^T R_Z(t - \tau) g_l(\tau) d\tau = s_l(t), \quad 1 \leq l \leq L, \quad 0 \leq t \leq T, \quad (9.43)$$

then (9.42) can be written as

$$R_l = \operatorname{Re} \left\{ \int_0^T Y(t) g_l^*(t) dt \right\} - \frac{1}{2} E_l, \quad E_l = \int_0^T s_l(t) g_l^*(t) dt. \quad (9.44)$$

**Example 9-19.**

If the additive noise is white,  $R_Z(\tau) = N_0 \delta(\tau)$ , then  $N_0 \cdot g_l(t) = s_l(t)$ . In that case, the receiver simply crosscorrelates with each of the known signals  $s_l(t)$ ,  $1 \leq l \leq L$ .  $\square$

The significance of (9.44) cannot be overstated. It shows that the infinite-dimensional continuous-time received signal can be reduced to a finite set of  $L$  decision variables  $R_l$ ,  $1 \leq l \leq L$ , where  $L$  is the number of known signals.  $R_l$  consists of a crosscorrelation against  $g_l(t)$ , as shown in Figure 9-7a. As in the discrete-time case, the correlation detector is equivalent to the continuous-time matched filter detector of Figure 9-7b. For the special case of white noise, Example 9-19 establishes that the matched filters in Figure 9-7 are matched to the set of signal waveforms  $s_l(t)$ ,  $1 \leq l \leq L$ .

If we let  $T \rightarrow \infty$ , we get a different interpretation and a better understanding of  $g_l(t)$ . In that case, assuming that  $g_l(t)$  is causal, (9.43) approaches a convolution equation  $R_Z(t) * g_l(t) = s_l(t)$ , or  $G_l(j\omega) = S_l(j\omega)/S_Z(j\omega)$ . This limiting case has the interpretation shown in Figure 9-7c. In the white noise case, the matched filter has impulse response  $s_l^*(t)$  and transfer function  $S_l^*(j\omega)$ .

In the nonwhite noise case, a different interpretation takes advantage of the fact that  $S_Z(j\omega)$  is positive real-valued. It can be factored as the product of two identical filters,

$$S_Z(j\omega) = S_Z^{1/2}(j\omega) \cdot S_Z^{1/2}(j\omega). \quad (9.45)$$

Thus, in Figure 9-7c the matched filter has been divided into two parts: a whitening filter  $1/S_Z^{1/2}(e^{j\omega T})$  that has white noise at the output, and a filter matched to the signal at the whitening filter output. The signal component at the output of the whitening filter is  $S_m(e^{j\omega T})/S_Z^{1/2}(e^{j\omega T})$ , and the second filter is matched to this new signal. The factorization into whitening and matched filtering is similar to the discrete-time case (Figure 9-6).  $E_l$  is the energy of the signal  $s_l(t)$  after it passes through the whitening filter, making it consistent with the white-noise case.

The factorization of (9.45) can be replaced by the product of two terms with equal magnitudes and arbitrary (rather than zero) phases, such that the product is real-valued. For example, we could replace (9.45) by a minimum-phase spectral factorization, similar to the factorization of (9.31) for the discrete-time power spectrum. This would have the benefit of explicitly controlling the causality or anticausality of the whitening and matched filters. However, since we are only using this factorization for intuition and not for realization, we will avoid this complication.

**9.3.3. Sufficient Statistics**

In (9.44), define the  $L$  complex-valued decision variables

$$V_l = \int_0^T Y(t) g_l^*(t) dt, \quad 1 \leq l \leq L, \quad (9.46)$$

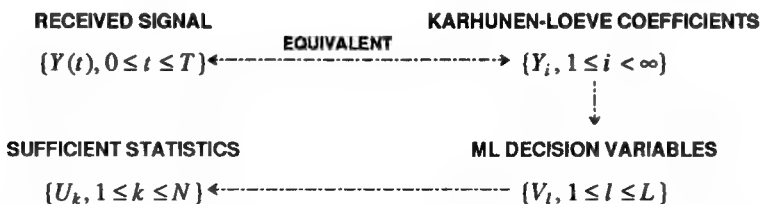
as labeled in Figure 9-7. The ML detector first calculates these  $L$  decision variables, corresponding to the  $L$  signals, and then chooses  $l$  to maximize  $R_l = \text{Re}\{V_l\} - E_l/2$ .

The ML detector is thus summarizing the continuous-time received signal  $\{Y(t), 0 \leq t \leq T\}$  by the  $L$  random variables  $\{V_l, 1 \leq l \leq L\}$ . In the process, it is clearly throwing away a lot of information about  $\{Y(t), 0 \leq t \leq T\}$ . The information that is being thrown away is considered *irrelevant* by the ML detector.

The overall goal of this subsection is summarized in Figure 9-8. Starting with the received signal  $\{Y(t), 0 \leq t \leq T\}$ , the Karhunen-Loeve expansion coefficients  $\{Y_i, 1 \leq i < \infty\}$  give an equivalent representation. This countable set of random variables is much easier to deal with analytically, but remains impractical for implementation because of the infinite number of variables. However, the ML detector further reduces the received signal to  $L$  ML decision variables  $\{V_l, 1 \leq l \leq L\}$ , where  $L$  is the number of known signals. For purposes of implementation, this finite set of decision variables is a dramatic improvement. Conceptually, however, if the dimensionality of the subspace spanned by the signals is less than  $L$ , then based on the experience of Chapter 8 we would expect that a number of decision variables equal to the dimension of the subspace would suffice. In fact, it will now be shown that the received signal can be represented by a set of  $N$  *sufficient statistics*  $\{U_k, 1 \leq k \leq N\}$ , where  $N \leq L$ , and  $N$  will be defined shortly. The sufficient statistics summarize  $\{Y(t), 0 \leq t \leq T\}$  for purposes of detection of  $\{s_l(t), 1 \leq l \leq L, 0 \leq t \leq T\}$ .

The  $N$  sufficient statistics can be used for purposes of ML detection. This reduces the number of decision variables that must be dealt with in the implementation of the ML detector. Remarkably, the sufficient statistics can be relied upon for the detection of the known signals  $\{s_l(t), 1 \leq l \leq L, 0 \leq t \leq T\}$  for *any* criterion of optimality, not just the ML criterion. For example, they would serve equally well as the starting point for MAP detection.

The intuitive basis for sufficient statistics is that they retain all the information in the received signal that is relevant to the detection of the known signals  $\{s_l(t), 1 \leq l \leq L, 0 \leq t \leq T\}$  and discard only information that is irrelevant. As we will also show below, the sufficient statistics  $\{U_k, 1 \leq k \leq N\}$  can be obtained from the ML decision variables  $\{V_l, 1 \leq l \leq L\}$  as pictured in Figure 9-8, and therefore all the information in the sufficient statistics must also be included in the ML decision variables. Thus, the ML decision variables are themselves sufficient statistics, and could be used by any detection criterion (not just the ML criterion) without



**Figure 9-8.** The progression from the received signal  $Y(t)$  through a progression of decision variables, all of which retain all relevant information for detection of the known signals.



compromising performance.

It will now be shown that  $\{U_k, 1 \leq k \leq N\}$  represent a sufficient statistic, and in the process make concrete the definition of sufficient statistic. This is done here for the limiting case of  $T \rightarrow \infty$ , and a more rigorous (and complicated) argument based on the Karhunen-Loeve expansion is given in Appendix 9-A. Returning to Figure 9-7c, let  $f_l(t)$  have Fourier transform

$$F_l(j\omega) = \frac{S_l(j\omega)}{S_Z^{1/2}(j\omega)}, \quad (9.47)$$

where  $S_l(j\omega)$  is the Fourier transform of the pulse  $s_l(t)$ . The signal  $f_l(t)$  is the response of the whitening filter  $1/S_Z^{1/2}(j\omega)$  to  $s_l(t)$ . Moreover, the second half of the matched filter in Figure 9-7c is matched to  $f_l(t)$ ; that is, it has impulse response  $f_l^*(-t)$ .

Assume the  $\{f_l(t), 0 \leq t < \infty, 1 \leq l \leq L\}$  span a subspace  $M_f$  of signal space of dimension  $N \leq L$ , and let  $\{\psi_k(t), 1 \leq k < \infty\}$  be a complete set of orthonormal functions chosen so that the first  $N$ ,  $\{\psi_k(t), 1 \leq k \leq N\}$ , serve as a basis for  $M_f$ . Then we can write

$$f_l(t) = \sum_{k=1}^N F_{l,k} \psi_k(t), \quad F_{l,k} = \int_0^\infty f_l(t) \psi_k^*(t) dt. \quad (9.48)$$

In the following, we will represent  $U(t)$ , the output of the whitening filter in Figure 9-7c, in terms of the basis  $\{\psi_k(t), 1 \leq k < \infty\}$ , and show that only the first  $N$  coordinates are relevant to detecting  $\{s_l(t), 1 \leq l \leq L, 0 \leq t \leq T\}$ .

The components of  $U(t)$  with respect to the new basis are

$$U_k = \int_0^\infty U(t) \psi_k^*(t) dt, \quad 1 \leq k \leq \infty. \quad (9.49)$$

Substituting for  $U(t)$  from

$$U(t) = f_l(t) + W(t) \quad (9.50)$$

where  $W(t)$  is white noise with unit variance,

$$\begin{aligned} U_k &= \int_0^\infty f_l(t) \psi_k^*(t) dt + \int_0^\infty W(t) \psi_k^*(t) dt \\ &= \begin{cases} F_{l,k} + W_k, & 1 \leq k \leq N \\ W_k, & N+1 \leq k < \infty \end{cases} \end{aligned} \quad (9.51)$$

The noise components  $W_k$  are mutually independent because of the white noise component in  $U(t)$  and the orthonormality of the  $\psi_k(t)$ . Only the first  $N$  components of  $U_k$  depend on the signal.

The  $\{U_k, 1 \leq k \leq N\}$  of (9.49) are sufficient statistics for the received signal  $Y(t)$ . This means they contain all the relevant information in  $\{Y(t), 0 \leq t \leq T\}$  for purposes of detection of known signals from the set  $\{s_l(t), 1 \leq l \leq L, 0 \leq t \leq T\}$  with

respect to *any* criterion of optimality, not just the ML criterion. In fact, the MAP detector, or any other detector can start by calculating these sufficient statistics. The justification for this sufficient statistic property is as follows:

- Only the first  $N$  components of  $U_k$  have a signal component.
- The remaining components are statistically independent of the first  $N$  components. Thus, they do not contain any information about the first  $N$  components.
- Thus  $\{U_k, N+1 \leq k < \infty\}$  is irrelevant to the detection of the signal, regardless of what criterion of optimality is used.

The procedure used to arrive at this smaller set of decision variables is identical to the expansion in Section 7.1, except there we expanded  $\{s_l(t), 1 \leq l \leq L, 0 \leq t \leq T\}$  in terms of an  $N$ -dimensional basis rather than the whitening-filter-adjusted signal set  $\{f_m(t), 1 \leq m \leq L\}$ . (Note that due to the whitening filter, the signal sets  $\{s_m(t), 1 \leq m \leq L\}$  and  $\{f_m(t), 1 \leq m \leq L\}$  may actually have a different dimensionality.) Thus, (9.48) and (7.6) are essentially the same, except for the intervening whitening filter.

#### Example 9-20.

As an example of the utility of the sufficient statistic argument, consider the reception of a single PAM pulse, where  $s_m(t) = A_m \cdot h(t)$  and the data symbol  $A_m$  assumes one of  $M$  values  $1 \leq m \leq M$ . For this case the received signal is one-dimensional, since all signals are linear multiples of a single waveform  $h(t)$ . Thus, if the additive noise is white and Gaussian, any detector can first form a sufficient statistic for the received signal

$$V_1 = \int_0^{\infty} Y(t)h^*(t) dt, \quad (9.52)$$

or if the noise is nonwhite, it may apply the received signal to a matched filter with transfer function  $H^*(j\omega)/S_Z(j\omega)$  and sample at time  $t = 0$ . The resulting *single* random variable summarizes the received signal for purposes of detection with respect to *any* criterion.  $\square$

It can be shown (Problem 9-23) that  $\{U_k, 1 \leq k \leq N\}$  can be obtained from the larger set of decision variables  $\{V_l, 1 \leq l \leq L\}$  by a simple linear transformation. This dependence is shown in Figure 9-8. It follows that  $\{V_l, 1 \leq l \leq L\}$  must also be a set of sufficient statistics, since  $\{U_k, 1 \leq k \leq N\}$  could not contain any relevant information about  $\{Y(t), 0 \leq t \leq T\}$  not present in  $\{V_l, 1 \leq l \leq L\}$ . It is usually advantageous to use  $\{U_k, 1 \leq k \leq N\}$  rather than  $\{V_l, 1 \leq l \leq L\}$  since it has fewer decision variables, and furthermore the Gaussian noise components in  $\{U_k, 1 \leq k \leq N\}$  are independent as established in (9.49). This latter property makes it easier to develop the remainder of the receiver structure based on the actual optimality criterion.

### 9.3.4. Applications of Sufficient Statistics

Based on the sufficient statistic results, and given a criterion of optimality, an optimal receiver structure can be developed as follows:

- The receiver front end calculates the  $N$  (or  $L$ ) sufficient statistics. One of these finite sets of decision variables replaces the received signal for purposes of the

design of the remainder of the receiver.

- The statistics of the sufficient statistics are established for the particular noise and the set of known signals.
- The remainder of the receiver structure is determined by the criterion of optimality as applied to the sufficient statistics and their statistical properties.

For stationary Gaussian noise, generally the receiver structures considered in Chapter 7 and analyzed in Chapter 8 are optimal with respect to the maximum-likelihood criterion. This statement is true with two qualifications: first, a whitening filter is required at the receiver front end and, second, the sufficient statistic argument of Appendix 9-A applies to a received signal over the finite interval  $[0, T]$ .

We will now establish that the minimum-distance receiver design of Chapter 7 is optimal with respect to the ML criterion, with the addition of an appropriate whitening filter.

### Sufficient Statistics for a Passband Signal

We now know that the front end (crosscorrelator or matched filter) of the receiver pictured in Figure 9-7 generates a set of sufficient statistics for the received signal. The remainder of the receiver is specialized to the ML criterion of optimality. We will now apply that result to the special case where the received signal is a real-valued passband signal. We expect that the sufficient statistics for this case will perform demodulation in addition to crosscorrelation or filtering. Because the noise is introduced on the passband channel, we have to start with the real-valued passband received signal rather than the complex baseband representation. Let the received signal be of the form

$$Y(t) = s_m(t) + N(t), \quad (9.53)$$

$$s_m(t) = \sqrt{2} \operatorname{Re} \{ \tilde{s}_m(t) e^{j\omega_c t} \}, \quad (9.54)$$

where  $\tilde{s}_m(t)$  is drawn from a set of  $L$  complex baseband signals, and  $N(t)$  is additive stationary real-valued Gaussian noise with autocorrelation  $R_N(\tau)$  and power spectral density  $S_N(j\omega)$ . Assume as well that  $g_m(t)$ , defined by (9.43), is written in terms of a complex baseband representation,

$$g_m(t) = \sqrt{2} \operatorname{Re} \{ \tilde{g}_m(t) e^{j\omega_c t} \}. \quad (9.55)$$

Then substituting (9.54) into (9.43),

$$\int_0^T R_N(t - \tau) \sqrt{2} \operatorname{Re} \{ \tilde{g}_m(\tau) e^{j\omega_c \tau} \} d\tau = \sqrt{2} \operatorname{Re} \{ \tilde{s}_m(t) e^{j\omega_c t} \}, \quad 0 \leq t \leq T. \quad (9.56)$$

Recognizing that  $R_N(\tau)$  is real valued, since  $N(t)$  is real valued, we can rewrite (9.56) as

$$\sqrt{2} \operatorname{Re} \{ e^{j\omega_c t} \int_0^T R_N(t - \tau) e^{-j\omega_c(t - \tau)} \tilde{g}_m(\tau) d\tau \} = \sqrt{2} \operatorname{Re} \{ \tilde{s}_m(t) e^{j\omega_c t} \}, \quad (9.57)$$

for  $0 \leq t \leq T$ . Thus, we can recast the integral equation in terms of baseband signals

as

$$\int_0^T R_N(t - \tau) e^{-j\omega_c(t - \tau)} \tilde{g}_m(\tau) d\tau = \tilde{s}_m(t), \quad 0 \leq t \leq T. \quad (9.58)$$

In particular, as  $T \rightarrow \infty$ ,

$$\tilde{G}_m(j\omega) \rightarrow \frac{\tilde{S}_m(j\omega)}{S_N[j(\omega + \omega_c)]}. \quad (9.59)$$

Recognizing that

$$\sqrt{2} \operatorname{Re}\{\tilde{g}_m(t) e^{j\omega_c t}\} = \sqrt{2} \operatorname{Re}\{\tilde{g}_m^*(t) e^{-j\omega_c t}\}, \quad (9.60)$$

the sufficient statistics become

$$\begin{aligned} V_m &= \int_0^\infty Y(t) g_m(t) dt = \int_0^\infty Y(t) \sqrt{2} \operatorname{Re}\{\tilde{g}_m^*(t) e^{-j\omega_c t}\} dt \\ &= \sqrt{2} \operatorname{Re}\left\{ \int_0^\infty Y(t) e^{-j\omega_c t} \tilde{g}_m^*(t) dt \right\}, \quad 1 \leq m \leq L. \end{aligned} \quad (9.61)$$

Another set of sufficient statistics for the received signal is clearly

$$\tilde{V}_m = \int_0^\infty Y(t) e^{-j\omega_c t} \tilde{g}_m^*(t) dt, \quad 1 \leq m \leq L. \quad (9.62)$$

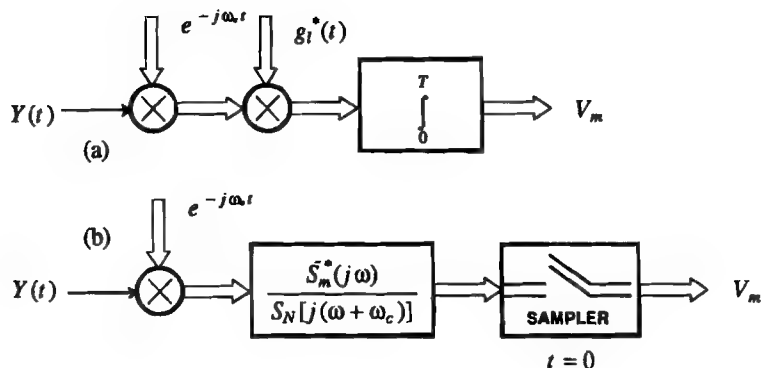
This is illustrated in Figure 9-9a, where the detector first demodulates and then crosscorrelates with the complex baseband waveform  $\tilde{g}_m^*(t)$ . An equivalent matched-filter realization is shown in Figure 9-9b. This is precisely the matched filter receiver considered in Chapter 7, except that the matched filter response is normalized by the power spectrum of the channel noise, translated from passband to d.c. because of the demodulator. When the noise is white, the structure specializes to precisely the receiver front end that arose out of the minimum-distance criterion in Chapter 7.

To generate fewer sufficient statistics, we can first whiten the received signal as shown in Figure 9-10, generating a new complex baseband received signal  $U(t)$  that contains complex-valued white Gaussian noise. A matched filter is then applied, where the transfer function  $\Psi_k(j\omega)$  is obtained as follows. First, ignoring the double-frequency and noise terms, the complex baseband signal at the output of the whitening filter has Fourier transform

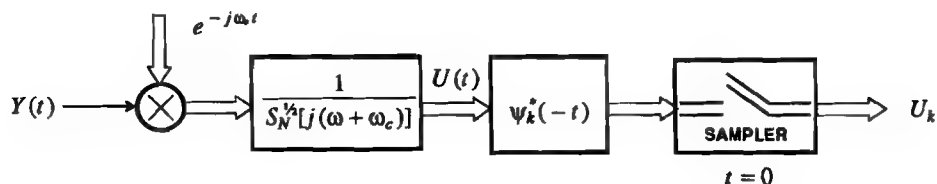
$$F_m(j\omega) = \frac{\tilde{S}_m(j\omega)}{S_N^{1/2}[j(\omega + \omega_c)]}. \quad (9.63)$$

Then an orthonormal basis  $\{\psi_k(t), 1 \leq k \leq N\}$  for the subspace spanned by  $\{f_m(t), 1 \leq m \leq L\}$  is chosen.

This result generalizes the ML detector results earlier, demonstrating that optimal detector structures for *all* criteria of optimality in additive stationary Gaussian noise can share a common receiver front end consisting of demodulator and a bank of



**Figure 9-9.** Generation of  $L$  sufficient statistics for a passband received signal  $Y(t)$ . (a) Cross-correlator and (b) matched filter realization, valid as  $T \rightarrow \infty$ .



**Figure 9-10.** A detector front-end that generates a set of  $N$  sufficient statistics for the passband received signal.

$L$  or  $N$  crosscorrelators or matched filters. Applying a given criterion is then a matter of characterizing the statistics of the resulting finite set of decision variables, and working out the optimal processing of those variables based on their statistics.

## 9.4. OPTIMAL INCOHERENT DETECTION

In Chapter 6, FSK was presented as a modulation technique suitable for transmission over channels that cause rapidly varying carrier phase. One of the major advantages of FSK is the ability to *incoherently* detect a signal, without deriving the carrier phase. Intuitively, this can be accomplished by realizing a set of bandpass filters, one centered at each of the known signal frequencies, and measuring the power at the output of each filter. The question arises, however, as to the optimal detection technique where the carrier phase is unknown. We will now derive the *optimal incoherent detector*, applying directly the results of Section 9.3.

Assume that the carrier phase is random, with the goal of rederiving the ML detector. The received signal is now of the form

$$Y(t) = \sqrt{2} \operatorname{Re} \{ \tilde{s}_m(t) e^{j(\omega_c t + \Theta)} \} + N(t), \quad 1 \leq m \leq L, \quad (9.64)$$

where  $\Theta$  is assumed independent of the signal, and the noise  $N(t)$  is white and Gaussian. In the absence of any other relevant information, we can assume that  $\Theta$  is uniformly distributed over the interval  $[0, 2\pi]$ ; this is also the most tractable choice analytically, leading to a simple result. The general approach to determining the ML detector is to first condition on knowledge of  $\Theta = \theta$ , and then average over  $\theta$ .

Assume that the  $\{\tilde{s}_m(t), 1 \leq m \leq L\}$  span a subspace of dimension  $N$ , and that this subspace has an orthonormal basis  $\{\phi_n(t), 1 \leq n \leq N\}$ . If the carrier phase is known to be  $\Theta = \theta$ , then a sufficient statistic is

$$\tilde{V}_n = \int_0^\infty Y(t) e^{-j\omega_c t} \phi_n^*(t) dt, \quad 1 \leq n \leq N, \quad (9.65)$$

as in (9.62). Incorporating phase  $\theta$  in the calculation of the sufficient statistic would simply multiply  $V_n$  by  $e^{j\theta}$ , but not add any additional information. Substituting (9.64) into (9.65), and observing that the  $2\omega_c$  term will integrate to zero, we can express the sufficient statistics as an  $N$ -dimensional vector,

$$\mathbf{V} = e^{j\theta} \tilde{\mathbf{S}}_m + \mathbf{Z}, \quad (9.66)$$

where  $\tilde{\mathbf{S}}_m$  is a vector of the coefficients of  $\tilde{s}_m(t)$  with respect to the orthonormal basis and  $\mathbf{Z}$  is a vector of independent circularly symmetric Gaussian random variables. The effect of the unknown carrier phase  $\theta$  is to shift the signal component of  $\mathbf{V}$  by phase  $\theta$ .

To determine the ML detector, we must determine the probability density function of  $\mathbf{V}$  conditioned on signal  $M$  being transmitted. As the first step, we find the p.d.f. of  $\mathbf{V}$  conditioned on both  $M$  and the phase  $\Theta$ ,  $f_{\mathbf{V}|M, \Theta}(\mathbf{v}|m, \theta)$ . This is a multi-dimensional Gaussian density function, given by

$$\begin{aligned} f_{\mathbf{V}|M, \Theta}(\mathbf{v}|m, \theta) &= \frac{1}{(2\pi)^N \sigma^{2N}} \exp \left[ -\frac{1}{2\sigma^2} \|\mathbf{v} - e^{j\theta} \tilde{\mathbf{S}}_m\|^2 \right] \\ &= \frac{1}{(2\pi)^N \sigma^{2N}} \exp \left[ -\frac{1}{2\sigma^2} (\|\mathbf{v}\|^2 + \|\tilde{\mathbf{S}}_m\|^2) \right] \exp \left[ \frac{1}{\sigma^2} \operatorname{Re} \{ e^{j\theta} \langle \mathbf{v}, \tilde{\mathbf{S}}_m \rangle^* \} \right], \end{aligned} \quad (9.67)$$

where  $\sigma^2 = N_0$  is the variance of the real or imaginary part of the Gaussian noise. This formidable expression will be made yet more formidable by finding  $f_{\mathbf{V}|M}(\mathbf{v}|m)$  by integrating out the dependence on  $\theta$ . But do not despair — the end result is simple! It is useful to derive first the following simple result.

#### Exercise 9-1.

Define the modified Bessel function of zero order,

$$I_0(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp\{x \cos(\theta)\} d\theta \quad (9.68)$$

for a real-valued  $x$ . Show that for a complex-valued  $z$ ,

$$I_0(|z|) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp\{ \operatorname{Re}\{ e^{j\theta} z^* \} \} d\theta. \quad (9.69)$$

**HINT:** Write  $z$  in polar coordinates.  $\square$

Using this result, we can find the marginal density of the received signal by integrating against the density function of  $\Theta$ ,

$$\begin{aligned} f_{\mathbf{v}|M}(\mathbf{v}|m) &= \int_{-\pi}^{\pi} f_{\mathbf{v}|M, \Theta}(\mathbf{v}|m, \theta) f_{\Theta}(\theta) d\theta \\ &= \frac{1}{(2\pi)^N \sigma^{2N}} \exp\left[ -\frac{1}{2\sigma^2} (\|\mathbf{v}\|^2 + \|\tilde{\mathbf{S}}_m\|^2) \right] I_0\left[ \frac{1}{\sigma^2} |\langle \mathbf{v}, \tilde{\mathbf{S}}_m \rangle| \right]. \end{aligned} \quad (9.70)$$

The final result will exploit a property of the Bessel function, that it is monotonic in its argument. Its precise shape is irrelevant.

The result is particularly simple when each signal has the same energy; that is, when  $\|\tilde{\mathbf{S}}_m\|$  is a constant. Then the exponential term in (9.70) is independent of  $m$ . From the monotonicity of  $I_0(x)$ , the ML receiver selects  $m$  to maximize

$$K_m = |\langle \mathbf{v}, \tilde{\mathbf{S}}_m \rangle| = \left| \sum_{n=1}^N v_n \tilde{s}_{m,n}^* \right| = \left| \int_{-\infty}^{\infty} Y(t) e^{-j\omega_c t} \tilde{s}_m^*(t) dt \right|. \quad (9.71)$$

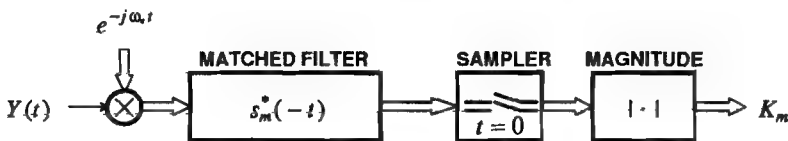
This is the simple form that was promised. A receiver structure to compute  $K_m$  is shown in Figure 9-11. Instead of correcting the matched filter output for the phase, as would be done if the phase were known, the receiver simply determines the magnitude of the matched filter output, throwing away any phase information.

#### Example 9-21.

For binary FSK,  $s_1(t) = e^{-j\omega_d t}$  and  $s_2(t) = e^{j\omega_d t}$ , where  $2\omega_d$  is the deviation between the two signals. The optimal receiver calculates the quantity

$$\left| \int_0^T Y(t) e^{-j(\omega_c \pm \omega_d)t} dt \right|^2 = \left[ \int_0^T Y(t) \cos(\omega_c \pm \omega_d)t dt \right]^2 + \left[ \int_0^T Y(t) \sin(\omega_c \pm \omega_d)t dt \right]^2 \quad (9.72)$$

as shown in Figure 9-12. Since the signal energies are the same, and since the Bessel



**Figure 9-11.** The optimal incoherent receiver uses a matched filter (a correlator could be used also) and throws away phase information.

function is monotonic,  $K_1$  and  $K_2$  given by (9.72) are compared and the signal corresponding to the maximum chosen. Intuitively, since the phase of the signal is unknown, we must correlate against two quadrature sinusoids, since we are then assured of a strong correlation for any signal phase for one or the other sinusoid phases. This receiver is also equivalent to passing the received signal through two filters

$$g(t) = \begin{cases} e^{j(\omega_c \pm \omega_d)t}, & -T \leq t \leq 0 \\ 0, & \text{otherwise} \end{cases} \quad (9.73)$$

These are roughly bandpass filters centered at  $\omega_c + \omega_d$  and  $\omega_c - \omega_d$ , the two transmitted frequencies. The filter outputs are each followed by an envelope detector. This structure was previously shown in Figure 6-48, where it was justified on intuitive grounds.  $\square$

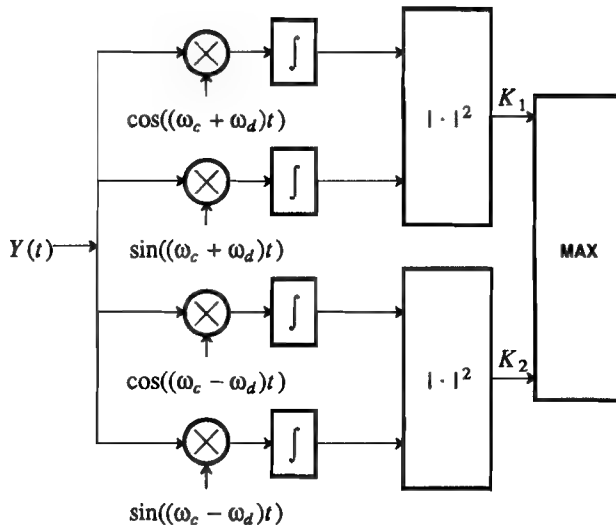
The calculation of the probability of error for an incoherent detector is rather involved, and each case is best treated individually. The starting point is substituting (9.66) into (9.71), so that the decision variable becomes, conditioned on transmitted signal  $l$ ,

$$K_m = \left| e^{j\theta} \langle \tilde{S}_l, \tilde{S}_m \rangle + \langle \mathbf{Z}, \tilde{S}_m \rangle \right|. \quad (9.74)$$

We can illustrate the probability of error calculation using FSK.

#### Example 9-22.

For FSK the two signals are orthogonal (with the proper choice of frequency deviation), and thus  $\langle \tilde{S}_1, \tilde{S}_2 \rangle = 0$ . Hence, the decision variables of (9.74) become, assuming  $\tilde{S}_1$  is transmitted,



**Figure 9-12.** The optimal incoherent receiver for a binary FSK signal using correlators.



$$K_1 = \left| e^{j\theta} \|\tilde{\mathbf{S}}_1\|^2 + \langle \mathbf{Z}, \tilde{\mathbf{S}}_1 \rangle \right|, \quad K_2 = \left| \langle \mathbf{Z}, \tilde{\mathbf{S}}_2 \rangle \right|. \quad (9.75)$$

The probability of error becomes, conditioned on  $\tilde{\mathbf{S}}_1$  transmitted,

$$P[\text{error} | \tilde{\mathbf{S}}_1 \text{ transmitted}] = \Pr\{ |\langle \mathbf{Z}, \tilde{\mathbf{S}}_2 \rangle| > |e^{j\theta} \|\tilde{\mathbf{S}}_1\|^2 + \langle \mathbf{Z}, \tilde{\mathbf{S}}_1 \rangle| \}. \quad (9.76)$$

This probability is in fact independent of  $\theta$ , and by symmetry is the same as the error probability conditioned on  $\tilde{\mathbf{S}}_2$  being transmitted. The two random variables on the left and right sides are not independent. The evaluation of the result is rather involved, and leads to an expression in terms of a tabulated function known as "Marcum's Q function".  $\square$

For more examples of the calculation of the probability of error, the interested reader is referred to [2,3].

## 9.5. OPTIMAL DETECTORS for PAM WITH ISI

In Chapter 7 a receiver for PAM with ISI was derived using a minimum-distance criterion. In Chapter 8, for white noise on the channel, the symbol rate noise samples at the output of the front-end filter were shown to be white. The front-end filter was therefore called a whitened matched filter (WMF). The variance of this noise was calculated, and the probability of a sequence error (one or more data symbols in error) was calculated.

In this section, we extend these results in several ways:

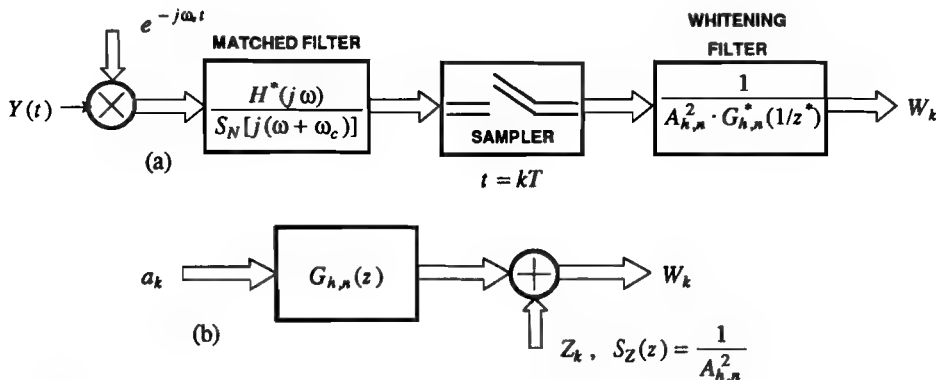
- We show that the WMF output is a set of sufficient statistics for the received signal. Thus, a receiver designed with respect to any criterion of optimality can share this same WMF front end.
- We show that the minimum-distance receiver design of Chapter 7 is, as expected, the ML detector for a set of  $M^K$  known signals consisting of a sequence of  $K$  data symbols. As a result, we call this receiver the *ML sequence detector (MLSD)*.
- We extend these results to nonwhite noise, and in particular show that the WMF can be reformulated for this case.

### WMF Outputs as Sufficient Statistics

As in Section 7.3, assume that the received signal consists of a finite sequence of  $K$  data symbols, each with the same alphabet with size  $M$ , modulating a basic complex baseband pulse  $h(t)$ ,

$$Y(t) = \sqrt{2} \operatorname{Re} \left\{ \sum_{k=1}^K a_k h(t - kT) e^{j\omega_c t} \right\} + N(t). \quad (9.77)$$

For the moment assume that the noise is white,  $S_N(j\omega) = N_0$ , as well as zero-mean and Gaussian. Then we can consider (9.77) as consisting of a signal portion drawn from a set of  $L = M^K$  known signals, together with additive white Gaussian noise. In Section 9.3, we established that a set of sufficient statistics can be generated by demodulating and correlating with the complex conjugate of each possible complex



**Figure 9-13.** A whitened matched filter for PAM and nonwhite noise. (a) Complete front end of the receiver, and (b) equivalent discrete-time model.

baseband signal,

$$\int_{-\infty}^{\infty} Y(t) \sum_{k=1}^K a_k^* h^*(t - kT) e^{-j\omega_c t} dt = \sum_{k=1}^K a_k^* U_k \quad (9.78)$$

where

$$U_k = \int_{-\infty}^{\infty} Y(t) \cdot e^{-j\omega_c t} h^*(t - kT) dt, \quad 1 \leq k \leq K. \quad (9.79)$$

This correlation has to be repeated for all  $L = M^K$  sequences of data symbols  $\{a_k, 1 \leq k \leq K\}$ .

In practice, calculating  $L = M^K$  correlations is not feasible as  $K$  gets large, but fortunately, (9.78) can be generated from the  $K$  decision variables  $U_k, 1 \leq k \leq K$  in (9.79). These  $K$  variables summarize the received signal from the perspective of calculating (9.78). The  $\{U_k, 1 \leq k \leq K\}$  are themselves sufficient statistics. This reduces the number of sufficient statistics from  $M^K$  down to just  $K$ . These  $K$  sufficient statistics are the outputs of  $K$  correlators against  $h^*(t - kT), 1 \leq k \leq K$ . As shown in Figure 7-9, these  $K$  correlators can be replaced by a single matched filter, matched to  $h(t)$ , followed by a sampler at  $t = kT$  for  $1 \leq k \leq K$ . Thus, we conclude that a receiver structure consisting of a demodulator followed by a filter matched to the complex baseband pulse  $h(t)$  followed by a symbol-rate sampler generates a set of sufficient statistics for the received signal detection.

This result is easily generalized to nonwhite noise using the results of Section 9.3, specifically the asymptotic results as  $T \rightarrow \infty$ . In this case, the output of the demodulator is first whitened by filter  $1/S_N^{1/2}[j(\omega + \omega_c)]$ , and the set of known signals is replaced by the set of known signals as modified by this whitening filter. Equivalently, the matched filter can be replaced by a filter matched to  $h(t)$  normalized by  $S_N[j(\omega + \omega_c)]$ . The resulting front end that generates the sufficient statistics  $\{U_k, 1 \leq k \leq K\}$  is shown in Figure 9-13a. The noise samples at the sampler output

are not white, but rather have power spectrum

$$S_{h,n}(e^{j\omega T}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} \frac{|H(j(\omega + m \cdot \frac{2\pi}{T}))|^2}{S_N[j(\omega + \omega_c + m \cdot \frac{2\pi}{T})]}. \quad (9.80)$$

This is similar to  $S_h(z)$  derived in Chapter 7, (7.36), except that  $H(j\omega)$  is normalized by  $S_N[j(\omega + \omega_c)]$ . As in Chapter 7, we can invoke a minimum-phase spectral factorization to write

$$S_{h,n}(z) = A_{h,n}^2 \cdot G_{h,n}(z) G_{h,n}^*(1/z^*) \quad (9.81)$$

where  $A_{h,n}^2$  is a positive constant and  $G_{h,n}(z)$  is a monic minimum-phase transfer function.

In Figure 9-13a a maximum-phase whitening filter is added to the output of the sampled matched filter, yielding an output noise process  $Z_k$  that is white, Gaussian, and circularly symmetric, with variance  $1/A_{h,n}^2$ . The resulting discrete-time channel model from input symbols to WMF outputs is shown in Figure 9-13b.

#### Example 9-23.

The front end of Figure 9-13 reduces to the WMF derived in Section 7.3 when the channel noise is white,  $S_N(j\omega) = N_0$ . For this case,  $S_{h,n}(e^{j\omega T}) = S_h(e^{j\omega T})/N_0$ , where  $S_h(e^{j\omega T})$  is the folded spectrum, and thus

$$S_{h,n}(z) = \frac{S_h(z)}{N_0} = \frac{A_h^2}{N_0} G_h(z) G_h^*(1/z^*). \quad (9.82)$$

The  $Z_k$  thus have variance  $1/A_{h,n}^2 = N_0/A_h^2$ , consistent with that determined in Section 8.6.  $\square$

A receiver for detection of the data symbols can be safely based on the WMF front end of Figure 9-13a regardless of what criterion of optimality is applied. This result is quite remarkable when we consider that symbol-rate sampling at the matched filter output is generally at less than the Nyquist rate, so aliasing of both noise and signal is inherent in this sampling. This aliasing will not compromise the performance of the receiver as long as the filter before the sampling is a matched filter. There are, however, practical concerns with this receiver structure that will be addressed in Chapter 10.

### Maximum-Likelihood Sequence Detector

The sufficient statistic argument allows us to use the front end of Figure 9-13a for any detection criterion, so we will now apply it to the ML detector. With the WMF front end, the equivalent discrete-time model of Figure 9-13b can be used as a starting point for application of the ML criterion. In particular, the ML detector for this equivalent discrete-time model was developed in Section 9.3. It chooses the sequence of data symbols that minimizes the Euclidean distance

$$\min_{\{a_k, 1 \leq k \leq K\}} \sum_{m=1}^{\infty} |W_m - \sum_{k=1}^K a_k g_{h,m-k}|^2. \quad (9.83)$$

This is precisely the minimum-distance receiver design of Chapter 7, and thus that receiver design is equivalent to the detector using the criterion of maximizing the likelihood of the received signal conditional on a sequence of data symbols  $\{a_k, 1 \leq k \leq K\}$ . Since an entire sequence of data symbols is detected at once, this detector is called the *maximum-likelihood sequence detector (MLSD)*. If all sequences are equally likely, the MLSD minimizes the probability of making one or more errors in a sequence of data symbols. That is, the criterion penalizes the detector equally for making any number of detection errors.

We have now derived the MLSD criterion of (9.83) in two ways. First, in Section 7.3 it was shown that the discrete-time criterion of (9.83) is equivalent to a continuous-time minimum-distance receiver design, in the sense that both criteria will choose the same sequence of data symbols. Second, in this section, using the argument that the WMF forms a sufficient statistic for the received signal detection, and also using the white noise property of the WMF output, we have shown that the criterion of (9.83) is optimal in the ML sense. Combining these two facts, we arrive at the conclusion that minimum-distance receiver design is optimal in the ML sense for PAM with ISI on the white Gaussian noise channel.

## 9.6. SEQUENCE DETECTION: THE VITERBI ALGORITHM

In Section 9.3 we derived the ML detector for the case where the reception is one of  $M$  signals. In many cases, the dimension  $N$  of the subspace of signals is too large to make the correlation (or equivalent matched filter) receiver practical. An example of this is the MLSD of Section 9.5, in which the signal corresponds to a sequence of  $K$  data symbols, and if they have alphabet size  $M$ , there are  $M^K$  total possible signals. The computation is therefore exponential in time (because time is proportional to  $K$ ). A practical realization requires that the computation be linear in time, since that corresponds to a fixed computational *rate*. More generally, the coding techniques covered in Chapters 13 and 14 use a signal set of high dimensionality as a way of combating noise, and would seem to suffer the same exponential dependence of computation on time.

Fortunately, there is a way to achieve a computational load that is linear in time in many practical situations, including coding as discussed in Chapters 13 and 14. Namely, we can impose some structure on the set of possible transmitted signals so that more sophisticated algorithms, equivalent to the correlator or matched filter but of much lower complexity, are possible. In particular, in this section we will assume a discrete-time model, and the signal-generation model is a finite-state machine (FSM). For this signal-generation model, the ML detector complexity can be reduced dramatically using a dynamic programming algorithm known as the *Viterbi algorithm*, originally proposed by A. Viterbi in 1967 [4].

### 9.6.1. Finite-State Machine Signal Generator

The Viterbi algorithm is applicable when the following properties hold:

- The signal is generated by a finite-state machine (FSM).
- The noise component in each sample is independent.
- A ML criterion is used, maximizing the sequence likelihood.

The output of an FSM driven by independent inputs is a homogeneous Markov chain (Section 3.3). This view of the signal generator output is a useful alternative viewpoint.

Let  $\Psi_k$  be the state sequence of a homogeneous Markov chain (Section 3.3). The sample space of each state  $\Psi_k$  is finite. For the signal generator of interest, the signal samples are a function of the Markov chain *state transitions*,

$$S_k = g(\Psi_k, \Psi_{k+1}), \quad (9.84)$$

where  $g(\cdot, \cdot)$  is a memoryless function. The objective of our ML and MAP detectors will be to detect the sequence of states given an observation sequence  $Y_k$ , which is  $S_k$  perturbed by independent noise components.

A special case that suits all our applications is the shift-register process of Example 3-13, reproduced in Figure 9-14 with a noise generator. Assume the input  $X_k$  is a sequence of i.i.d. random variables with a finite sample space. The state of the Markov chain is  $\Psi_k = [X_{k-1}, X_{k-2}, \dots, X_{k-J}]$ , where  $J$  is the length of the shift-register.

#### Example 9-24.

When the received signal is PAM with ISI, and the front-end filter is the WFM of Figure 9-13a, the resulting discrete-time channel model of Figure 9-13b is in the form of a signal-generation model followed by a noise generation model. The signal generation model is a filter  $G_{h,n}(z)$  driven by the data symbols  $a_k$ , and the noise generation model is additive complex-valued Gaussian noise with independent samples. The signal-generation model is in general not an FSM, unless the filter is FIR. If  $G_{h,n}(z)$  is FIR,

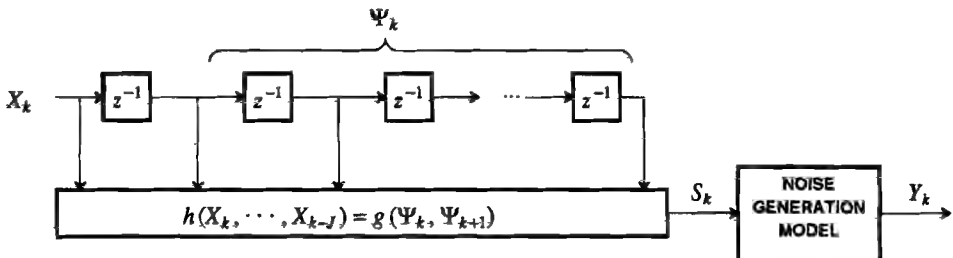


Figure 9-14. A shift-register process with an observation function and noise generator.

$$G_{h,n}(z) = \sum_{k=0}^J g_{h,k} z^{-k}, \quad (9.85)$$

then the FSM has state

$$\Psi_k = [a_{k-1}, a_{k-2}, \dots, a_{k-J}]. \quad (9.86)$$

( $G_{h,n}(z)$  is FIR if and only if  $S_{h,n}(z)$  is an all-zero filter, or equivalently only a finite set of translates  $h(t - kT)$  after whitening are non-orthogonal.) There are then  $M^J$  if the signal alphabet has size  $M$ . This ISI channel model is an example of a shift-register process, where the observation function is

$$g(\Psi_k, \Psi_{k+1}) = \sum_{i=0}^J g_{h,i} a_{k-i}. \quad (9.87)$$

In addition, the independent noise samples at the output of the WMF satisfy the assumptions required for application of the Viterbi algorithm.  $\square$

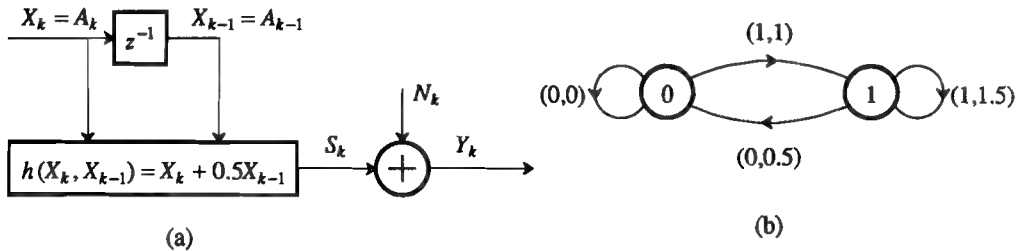
### Example 9-25.

A simple example that we will carry along for illustrative purposes is the ISI model

$$g_k = \delta_k + 0.5\delta_{k-1}, \quad (9.88)$$

as shown in Figure 9-15. Using the notation of Figure 9-14, this is the shift-register process shown in Figure 9-15a. If the input symbols are i.i.d., the observation  $Y_k$  is a noisy observation of the Markov chain with state transition diagram shown in Figure 9-15b. We have assumed binary inputs  $A_k$ , so that there are only two states, corresponding to the two possible values for  $A_{k-1}$ . The arcs are labeled with the input/output pair  $(A_k, S_k)$  of the signal generator.  $\square$

The Markov chain signal generator, and the shift-register process in particular, also model an important coding technique considered in Chapter 13. Although it is premature to talk about coding here, we will nevertheless present this as another example of a signal generation model.

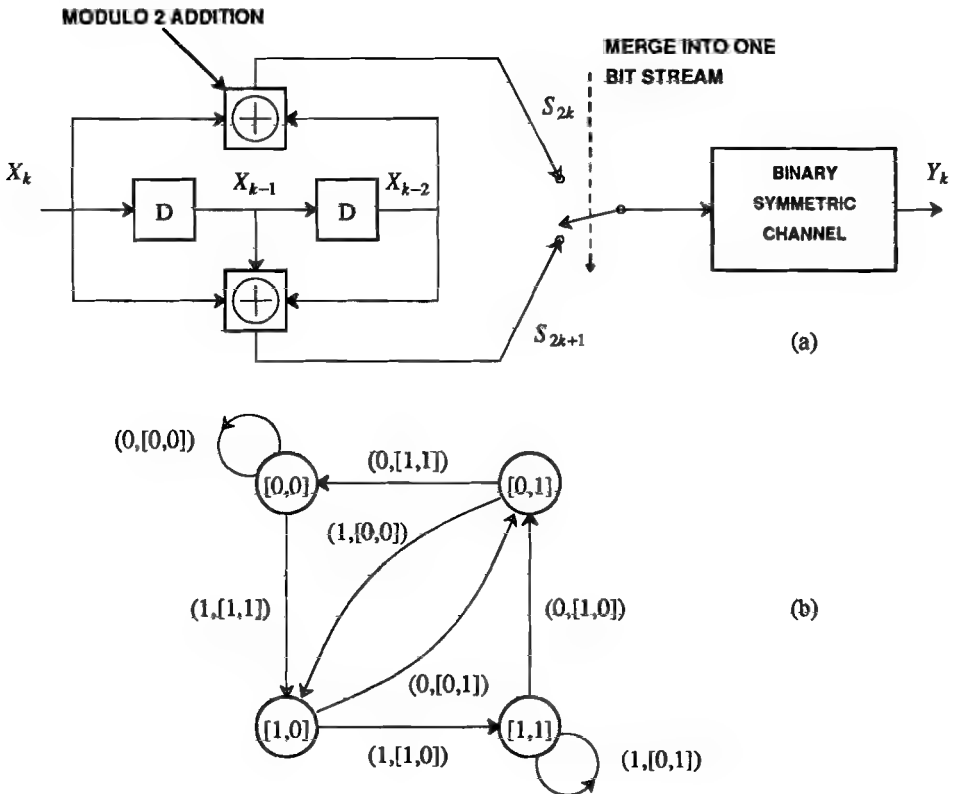


**Figure 9-15.** a. The ISI signal generator of Example 9-25 is a shift-register process, which is a Markov chain when the data symbols  $A_k$  are i.i.d. b. The state transition diagram for the Markov signal generator assuming binary input symbols. The arcs are labeled with the input bit/signal output pair  $(A_k, S_k)$ .

**Example 9-26.**

*Convolutional coders* introduce redundancy so that random errors occurring in the transmission of a data sequence can be corrected. An example of a convolutional coder is shown in Figure 9-16a. In this case the input  $X_k$  is binary, usually the bits to be transmitted. The signal  $S_k$  is a binary channel input ( $S_k \in \{0,1\}$ ). Two symbols are generated for each data bit  $X_k$ , namely  $S_{2k}$  and  $S_{2k+1}$ , so that  $S_k$  are actually transmitted at twice the bit rate of the input bit stream  $X_k$ . The noise generator often assumed in this case is the BSC with independent noise components. Assuming that the input sequence is i.i.d., we can define the state of a Markov chain to be  $\Psi_k = [X_{k-1}, X_{k-2}]$ . The state transition diagram is shown in Figure 9-16b. There are four states corresponding to the two past input bits  $X_{k-1}$  and  $X_{k-2}$ .  $\square$

One of the characteristics of the output  $S_k$  produced by the Markov signal generator is redundancy. In Example 9-25, the redundancy occurs because  $S_k$  takes on four possible levels (0.0,0.5,1.0,1.5) even though only one bit of information is carried. The ISI which introduces this redundancy is assumed to be an undesired



**Figure 9-16.** a. A rate  $\frac{1}{2}$  convolutional coder feeding a binary symmetric channel (BSC), which randomly (with probability  $p$ ) inverts bits. b. The state transition diagram. The arcs are labeled with the input bit and the pair of output bits  $(X_k, [S_{2k}, S_{2k+1}])$ .

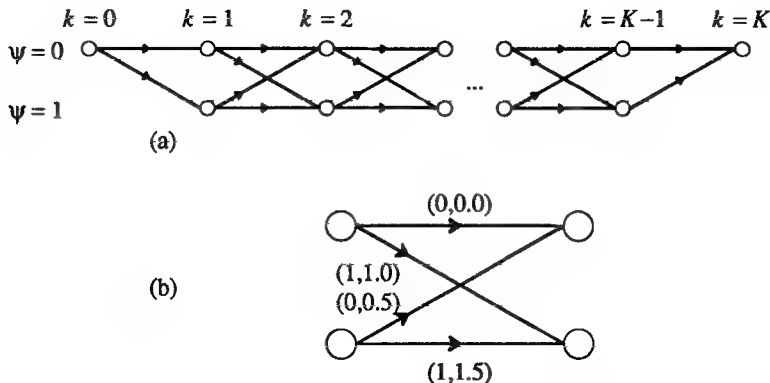
property of the channel, although there are situations where it is introduced deliberately — for example with *partial response* in Chapter 12. In Example 9-26 the redundancy occurs because two bits are transmitted through the BSC for every bit of information, with the goal of mitigating the effect of errors introduced on the BSC. Although the form and function of the convolutional coder is very different from the ISI channel, the principles of sequence detection developed in this section apply equally to both cases. In the remainder of this section we will discuss the technique of sequence detection, and leave detailed discussion of its applications to Chapters 10, 12, 13, and 14.

### 9.6.2. The Trellis Diagram

The state transition diagrams in Example 9-25 and Example 9-26 are traditional representations of Markov chains. D. Forney suggested in 1967 a valuable alternative representation called a *trellis diagram* [5], which shows the possible progression of states over time.

#### Example 9-27.

The state transitions of Example 9-25 are shown in the trellis diagram in Figure 9-17a, subject to the starting and ending conditions  $\Psi_0 = 0$  and  $\Psi_K = 0$ . Each small circle is a *node* of the trellis, and corresponds to the Markov chain being in a particular state at a particular time. Each arc in the diagram is called a *branch*, and corresponds to a particular state transition at a particular time. Thus, the single node at the left indicates that the Markov chain begins in state  $\psi_0 = 0$  at time  $k = 0$ . The next state can be either state  $\psi_1 = 0$  or state  $\psi_1 = 1$ , so transitions to both are shown. After time  $k = 1$ , the Markov chain for this example may branch (transition) from any node (state) to any other node (state), until it reaches the terminal node of the trellis in state  $\psi_K = 0$ . Each branch in the trellis corresponds to one state transition that is triggered by a particular input  $X_k$  and produces the output  $S_k$ , and thus there is a one-to-one correspondence at time  $k$  between a branch, the state transition, and



**Figure 9-17.** (a) A two-state trellis illustrating the possible state transitions of the Markov chain in Example 9-25, assuming the initial and final states are zero. (b) One stage of the two-state trellis is shown labeled with the input and output pairs  $(X_k, S_k)$  corresponding to the state transition.



both the input and output of the signal generator. One segment of the trellis is shown in Example 9-27b with the input and output pairs  $(X_k, S_k)$  labeled for each transition.  $\square$

A sequence of branches through the trellis diagram from the beginning to terminal nodes is called a *path*. Every possible path corresponds to an input sequence  $X_k, 0 \leq k \leq K$ . The goal of a detector, based on the observation of  $S_k$  corrupted by noise, is to decide on the sequence of inputs. Deciding on a sequence of inputs is equivalent to deciding on a path through the trellis diagram. The detector in this case is called a *sequence detector*, since it is simultaneously deciding on an entire sequence of inputs (or a path through the trellis) rather than deciding on one input at a time.

### 9.6.3. ML and MAP Sequence Detectors

Our goal is to design a MAP or ML detector for the state sequence  $\Psi_k$ , input  $X_k$ , output signal  $S_k$ , or path through the trellis diagram (all of which are equivalent), based on the noisy observation sequence  $Y_k$ . In principle we have already solved this problem, since this is an example of the vector signal generation model.

#### Example 9-28.

In the Gaussian noise case the ML detector chooses the signal  $\hat{s}_k$  that minimizes the squared Euclidean distance between the observation and the signal,  $\sum_{k=0}^K |y_k - \hat{s}_k|^2$ , where  $y_k$  is the observed outcome of the random variable  $Y_k$ .  $\square$

#### Example 9-29.

The ML detector for the BSC minimizes the Hamming distance  $\sum_{k=0}^K d_H(y_k, \hat{s}_k)$ , where  $d_H(u, v)$  is the Hamming distance between  $u$  and  $v$ . (For the convolutional coder of Figure 9-16, there are actually  $2K+2$  bits generated for  $K+1$  input bits, so the upper limit of the summation should be  $2K+1$ .)  $\square$

We can relate these results back to the trellis diagram. Recall that there is a signal  $s_k$  associated with each branch of the trellis at each stage  $k$  of the trellis. For each stage  $k$  there is also an observation  $y_k$ . After observing  $y_k$ , we can assign to each branch of the trellis a numerical value called the *branch metric* that is low if  $y_k$  is close to  $s_k$  and high otherwise. For the Gaussian case the appropriate branch metric is

$$\text{branch metric} = |y_k - s_k|^2, \quad (9.89)$$

and for the BSC it is

$$\text{branch metric} = d_H(y_k, s_k). \quad (9.90)$$

Then for each path through the trellis, we can calculate the *path metric*, which is the sum of the branch metrics. The preferred path will be the one with the lowest path metric.

In Appendix 9-B we generalize this to any noise generator with independent noise components, and also generalize to the MAP detector. In each case the objective is to minimize a path metric that is the sum of branch metrics, where the only

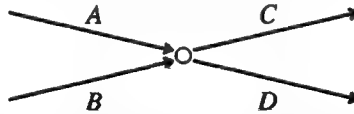
difference between the detectors is the formula for the branch metric.

The ML or MAP detector first calculates the branch metrics for every branch in the trellis diagram. It then calculates the path metric for every path in the trellis diagram, and chooses the path for which this path metric is minimum. The detected input sequence is then the sequence corresponding to this path. This straightforward approach of exhaustively calculating the path metric for each and every path through the trellis will clearly fail in practice because the number of paths grows exponentially with  $K$ . Usually  $K$  will be very large, corresponding to the entire time that communication takes place (usually minutes, hours, or even decades!). The Viterbi algorithm is a computationally efficient algorithm that exploits the special structure of the trellis to achieve a complexity that grows only linearly with  $K$ , or in other words, requires a constant computation rate (per unit time).

Consider one node in the trellis diagram, and all paths through the trellis that pass through this node.

#### Example 9-30.

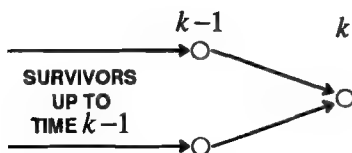
The particular case of two incoming branches and two outgoing branches is shown below:



The incoming branches are labeled  $A$  and  $B$ , and the outgoing branches are labeled  $C$  and  $D$ . There are a large number of paths passing through this node (increasing exponentially with  $K$ ), but all these paths follow one of just four routes through this node,  $AC$ ,  $AD$ ,  $BC$ , and  $BD$ .  $\square$

The path metric for a particular path through the node is the sum of the *partial path metrics* for the portion of the path to the left and the portion to the right of the node. Among all possible partial paths to the left, the detector will always prefer the one with the smallest partial path metric, called the *survivor path* for that node. We can immediately remove from consideration all partial paths to the left other than the survivor path, because any other partial path to the left has by definition a larger partial path metric, and if it replaced the survivor path in any overall path, the path metric would be larger. This is the basis of the Viterbi algorithm, which allows us to reject many possible paths at each stage of the trellis.

The Viterbi algorithm finds the path with the minimum path metric by sequentially moving through the trellis and at each node retaining only the survivor paths. At each stage of the trellis we do not know which node the optimal path passes through, so we must retain one survivor path for each node. When we reach the terminal node of the trellis, we find the optimal path, which is the single survivor path for that node. The algorithm thus determines, at each time increment  $k$ , the survivor path for each of the  $N$  nodes. The trick, then, is finding these  $N$  survivor paths based on the information developed up to time  $k - 1$ . This is pictured below for the case where there are two incoming branches to a given node at time  $k$ :



The only incoming paths to a node at time  $k$  that are candidates to be survivors are those consisting of survivors at time  $k - 1$  followed by branches to time  $k$ . (The number of such candidates is equal to the number of incoming branches to that node.) We therefore determine the partial path metrics for each of those candidate paths by summing the partial path metric of the survivor at time  $k - 1$  and the metric of the branch to time  $k$ . The survivor path at time  $k$  for a given node is the candidate path terminating on that node with the smallest partial path metric. We must store, for each node at time  $k$ , the survivor path and the associated partial path metric, for the algorithm to proceed to time  $k + 1$ . We will illustrate the Viterbi algorithm with an example.

#### Example 9-31.

The trellis shown in Figure 9-18 is marked with branch metrics corresponding to the observation sequence  $\{0.2, 0.6, 0.9, 0.1\}$  for the additive Gaussian noise case of Example 9-25. The path metrics  $|y_k - s_k|^2$  are labeled in Figure 9-18. A simple ML slicer (not a sequence detector) would decide that the transmitted bits were  $\{0, 1, 1, 0\}$ , but the ML sequence detector takes into account knowledge of the ISI and selects  $\{0, 1, 0, 0\}$ . An iterative procedure for making this decision is illustrated in Figure 9-18. The survivor paths at each node and the partial path metric of each surviving path are shown.  $\square$

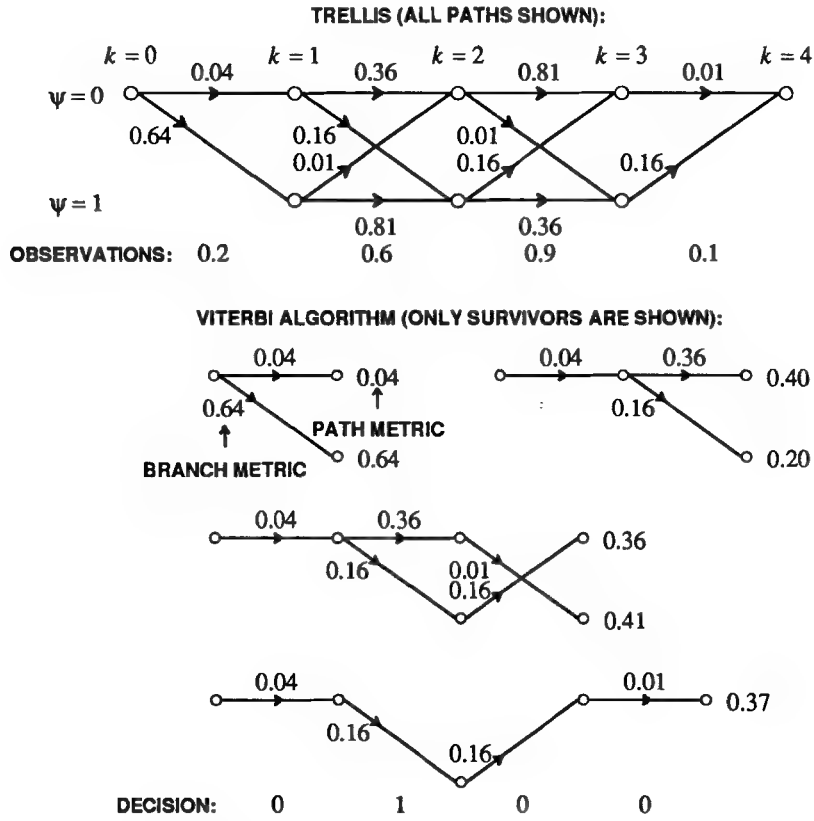
The computational complexity of the Viterbi algorithm is the same at each time increment, except for end effects at the originating and terminating nodes, and hence the total computational complexity is proportional to the length of time  $K$ .

One practical problem remains. The algorithm does not determine the optimal path until the terminal node of the trellis; that is, it does not reach a conclusion on the entire ML or MAP sequence until the end of the sequence. Further, while the computation at each step is the same, the memory required to store the survivor paths grows linearly with time. In digital communications systems, sequences may be very long, and we cannot afford the resulting long delay in making decisions nor the very large memory that would be required. It is helpful if at some iteration  $k$ , all the survivor paths up to iteration  $k - d$  coincide, for some  $d$ .

#### Example 9-32.

In Example 9-31, when  $k \geq 2$ , all the survivor paths coincide from  $k = 0$  to  $k = 1$ . The ML or MAP detector decision for the first state transition can be made when  $k = 2$ . It is not necessary to wait until the terminal node of the trellis.  $\square$

When all the survivor paths at some time  $k$  coincide up to some time  $k - d$ , we say that the partial paths have *merged* at depth  $d$ , and we can make a decision on all the inputs or states up to time  $k - d$ . Unfortunately, we cannot depend on the good fortune of a merge, as it is possible for no merges to occur. It is usual therefore to make a modification to the algorithm by forcing a decision at time  $k$  on all transitions prior



**Figure 9-18.** A two-state trellis with the branch metrics of the transitions marked and the Viterbi algorithm illustrated. The Viterbi algorithm iteratively finds the path with the minimum path metric without ever considering more than two paths at once.

to time  $k - d$ , for some *truncation depth*  $d$ . The usual approach is to compare all the partial path metrics for the  $N$  partial paths at time  $k$ , and note which one is the smallest. The decision on the input or state transition at time  $k - d$  is then the transition at  $k - d$  of this survivor path. Since the decision has been made, there is no need to store the survivor paths beyond a depth of  $d$  transitions in that path. If  $d$  is chosen to be large enough, this modification will have negligible impact on the probability of detecting the correct sequence.

**9.6.4. Error Probability Calculation**

We have already analyzed in Section 9.2 the probability of error in a vector detection problem. If the entire sequence from  $k = 0$  to  $k = K$  is considered to be a vector, then the result in Section 9.2 can be applied directly. The fact that we have found a computationally efficient algorithm for making the ML decision will not change that error probability. The *sequence error probability* is the probability that

the path chosen through the trellis does not correspond to the correct state sequence, or in other words the probability that *one or more* detected states are in error. This criterion thus gives the same weighting to an error in which ten states are incorrect as when only one state is in error. We showed in Section 9.2 that this error probability is dominated by the path through the trellis that is nearest in distance (Euclidean or Hamming) to the correct path. However, caution is in order! As the length  $K$  of the sequence gets large, the number of distinct paths with minimum distance from the correct path also gets large, invalidating the approximations in Section 9.2. In fact, as  $K$  gets large, the probability of sequence error usually approaches unity!

The usual measure of performance for a digital communication system, however, is the probability of a single symbol or bit error, or the probability of a sequence error for a relatively short sequence (such as one block of data). This observation has two implications. First, for many applications the ML sequence criterion is not the appropriate one. It might be more appropriate to minimize the data symbol or bit error probability instead. Second, even if we use the sequence detector, we would like to know the probability of a bit or symbol error, and the relationship of these to the sequence error probability is not trivial. A simple approach, developed below, is to calculate the probability of a sequence error per unit time, and then relate this normalized sequence-error probability to the probability of a symbol and bit error.

In practice the ML sequence criterion is usually used in preference to minimizing the probability of a symbol error, for two reasons. First, it is much simpler to implement, and in fact the best known algorithms for optimal bit-by-bit detection have exponential complexity in  $K$ . Second, the performance of the ML sequence detector is almost identical at high SNR (Gaussian noise) or low channel error probability (BSC) to the detector that minimizes the bit or symbol error probability.

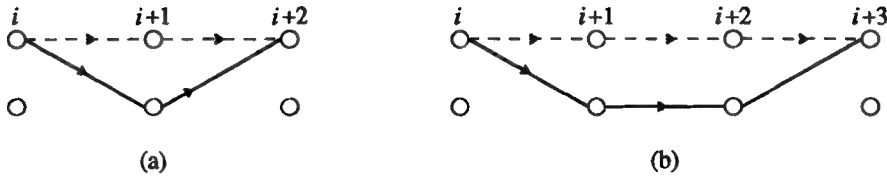
## Error Events

For purposes of determining the symbol error probability, it is useful to introduce the concept of an *error event*. Let  $\{\psi_k\}$  be the correct state sequence and  $\{\hat{\psi}_k\}$  be the sequence selected by the Viterbi algorithm. Over a long time,  $\{\psi_k\}$  and  $\{\hat{\psi}_k\}$  will typically diverge and remerge several times. Each distinct separation is called an error event, which is therefore defined as a correct path through the trellis paired with an error path that begins and ends with the correct state. By definition, the error path does not share any intermediate states with the correct state sequence. The *length* of an error event is the number of intermediate (incorrect) nodes in the path.

### Example 9-33.

Examples of error events of length one and two are shown in Example 9-33 for a two-state trellis. The assumed correct state trajectory is shown by dashed lines, and the error event by solid lines. There are error events of unbounded length, although as we will see, the probability of the longer events will usually (but not always) be negligibly small.  $\square$

An error event has one or more *symbol errors*, which are incorrect symbols or bits that result from taking an incorrect path through the trellis. In Appendix 9-C we show that the probability of symbol error is dominated by the probability of the minimum distance error event at high SNR. For the Gaussian noise case,



**Figure 9-19.** When the correct state sequence  $\psi$  and the detected state sequence  $\hat{\psi}$  diverge and remerge, we have an error event. Two error events are shown here. (a) The shortest error event for the two-state trellis in Example 9-25. (b) The next longest error event. In both cases we have assumed the correct state trajectory is all zeros, shown with the dashed lines.

$$\Pr[\text{symbol error}] \approx C \cdot Q(d_{\min}/2\sigma) \quad (9.91)$$

and for the BSC case

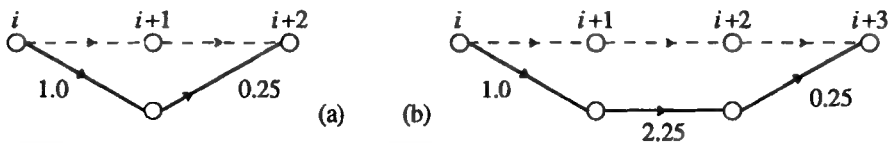
$$\Pr[\text{symbol error}] \approx C \cdot Q(d_{\min}, p), \quad (9.92)$$

where  $C$  is some constant between  $P$  and  $R$  given by (9.158) and (9.150) respectively, and  $Q(\cdot, \cdot)$  is defined by (9.25). As long as  $P$  and  $R$  are reasonably close to unity, we need not be too concerned with this multiplicative constant.

The following procedure will find the distance of any particular error event for either the Gaussian or BSC cases. Assume a correct state sequence, and label each branch in the trellis with its squared distance from the corresponding branch of the correct state sequence. This would be the branch metric if the channel were noiseless. The correct state sequence will have branch metrics that are zero, and normally all the branches not on the correct path will have a non-zero branch metric. For each possible error event, we can find the distance of that error event very simply by computing its path metric.

#### Example 9-34.

Continuing the ISI Example 9-25, Figure 9-20 shows the trellis labeled with the branch metrics assuming a noiseless channel and an all-zeros transmitted sequence. The path metric for each path through the trellis is now the square of the Euclidean distance of that path from the correct all-zeros path. The error event of length one is easily seen to have



**Figure 9-20.** The trellis of Figure 9-19 labeled with the distances from the correct branches, assuming the correct branches are the dashed ones.

Euclidean distance  $\sqrt{1.25}$ , and the error event of length two has distance  $\sqrt{3.5}$ . Longer error events have still greater distances. Obviously, the error event of length one is much more probable than longer error events. It is easy to show by exhaustive search that all possible correct paths through the trellis are at least distance  $\sqrt{1.25}$ , and none has smaller distance (see Exercise 9-2 below), so  $\sqrt{1.25}$  is the minimum distance for all possible correct paths. It is shown in Appendix 9-C that  $C = 1$ , so

$$\Pr[\text{symbol error}] \approx Q(\sqrt{1.25}/2\sigma). \quad (9.93)$$

□

### Exercise 9-2.

Completing Example 9-34, show that for each possible correct path through the trellis, the minimum-distance error event has length one and distance  $\sqrt{1.25}$ . □

In Example 9-34 we found the minimum distance by inspection. This will not be so easy in general. Fortunately, it turns out that the Viterbi algorithm can itself be used to find the minimum-distance error event for a given correct state sequence! Using the trellis diagram labeled with the same branch metrics as above, we can begin on the correct path and use the Viterbi algorithm to find the survivor at each stage, excluding the correct path (the only one with zero path metric). Each survivor at a node on the correct path corresponds to an error event, and the path metric is the distance of this error event from the correct path. At each stage of the algorithm, keep track of the minimum-distance error events recorded thus far; when all survivors have a partial path metric greater than this minimum, the minimum-distance error event has been found for one assumed correct path through the trellis.

As shown in Appendix 9-C, it is the global minimum distance  $d_{\min}$  that dominates the probability of error, not the minimum distance from one particular path through the trellis. Fortunately, usually we do not need to examine all possible correct paths to find the minimum distance. Better techniques are available for both the ISI examples (discussed shortly), and the BSC examples (discussed in Chapter 13). In both cases we exploit symmetry based on linearity, although the nature of the linearity is quite different in each case.

## 9.6.5. Calculating the Minimum-Distance for ISI

The performance of the MLSD is determined largely by the minimum-distance properties of the ISI. This can be determined in brute force fashion by finding the minimum-distance error event for all possible starting paths through the trellis. However, due to the linearity of the FSM model in the particular case of ISI, the problem can be simplified. In particular, if we define an *error symbol*  $\epsilon_k$  as  $\epsilon_k = a_k - \tilde{a}_k$  where  $a_k$  and  $\tilde{a}_k$  are feasible data symbols, then it was shown in (7.72) that the minimum distance is given by

$$d_{\min}^2 = \min_{\{\epsilon_k, 1 \leq k \leq K\}} \sum_{m=1}^{\infty} \left| \sum_{k=1}^K \epsilon_k g_{h,m-k} \right|^2, \quad (9.94)$$

where the minimization is over all *non-zero* sequences of error symbols; that is, at

least one of the error symbols must be non-zero. By time invariance, we can limit attention to error events that begin with an error at time  $k = 1$ ; that is, the minimization of (9.94) is over all  $\{\epsilon_k, 1 \leq k \leq K\}$  such that  $\epsilon_1 \neq 0$ .

The minimum-distance problem of (9.94) can be formulated in a form that can be solved by the Viterbi algorithm, but only for the case where  $G_h(z)$  is an FIR filter, where an FSM model holds. Assuming  $g_{h,k} = 0$  for  $k > J$ , the convolution sum can be reversed,

$$d_{\min}^2 = \min_{\{\epsilon_k, 1 \leq k \leq K\}} \sum_{m=1}^{K+J} \left| \sum_{i=0}^J g_{h,i} \epsilon_{m-i} \right|^2. \quad (9.95)$$

Two desirable things happen in the FIR case as formulated in (9.95). First, the summation over  $m$  becomes finite, although in practice this is not too helpful because we are interested in very large values of  $K$ . Second, the minimization can be formulated as the minimization of the path metric in a trellis. Toward this end, define a state

$$\Psi_k = [\epsilon_{k-1}, \dots, \epsilon_{k-2}, \epsilon_{k-J}] \quad (9.96)$$

and a trellis diagram for the progression of this state. Labeling each branch of this trellis with the corresponding branch metric  $\left| \sum_{i=0}^J g_{h,i} \epsilon_{k-i} \right|^2$ , the minimization problem

becomes one of finding the non-zero path through this trellis with minimum path metric. The minimization is over all error symbol sequences starting with  $\epsilon_1 \neq 0$ , and hence over all paths through the trellis where the starting state is  $[0, 0, \dots, 0]$  and the first branch is non-zero.

This formulation of the minimum-distance problem has its price, as well as a major advantage. The price is that the alphabet of error symbols is larger than the alphabet of the data symbols, increasing the number of states in the trellis. In general if the data symbol alphabet has size  $M$ , the error symbol alphabet can be as large as  $M^2$ , although normally it is smaller.

#### Example 9-35.

If the data symbols are real-valued,  $M$  is odd, and the data-symbol alphabet is all integers in the range  $[-(M-1)/2, (M-1)/2]$ , then the error symbols (difference between two data symbols) can assume all values in the range  $[-(M-1), (M-1)]$ , or  $(2M-1)$  distinct values. This is generally much smaller than  $M^2$ .  $\square$

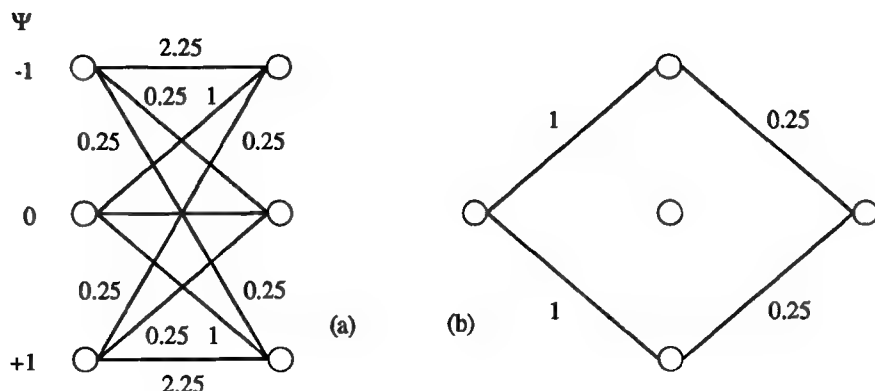
The advantage of this formulation is that the branch metric is a function of each branch in isolation, not pairs of branches (correct path and error path), greatly reducing the number of alternatives that have to be considered.

The Viterbi algorithm can now be applied to finding  $d_{\min}$  in a simple example (where it is hardly needed).

#### Example 9-36.

For the ISI channel of Example 9-25, where  $G(z) = 1 + 0.5z^{-1}$ , the branch metric is  $(\epsilon_k + 0.5\epsilon_{k-1})^2$ . If the data symbols are binary, with alphabet  $\{0, 1\}$ , the error  $\epsilon_k$  is ternary, with alphabet  $\{0, \pm 1\}$ . The corresponding three-state trellis diagram is shown in Figure 9-21a, and the two minimum-distance error events are shown in Figure 9-21b. Each error





**Figure 9-21.** a. The trellis diagram for finding the minimum distance for the ISI example of Example 9-36. b. The two minimum-distance error events.

event has a single symbol error.  $\square$

To complete the probability-of-error analysis it is helpful to estimate the error coefficient  $C$  in (9.91) or (9.92) by finding  $P$  and  $R$ , where  $P \leq C \leq R$ . Recall that  $P$  is the probability that there is an error event starting at a fixed time  $i$  with distance  $d_{\min}$ , and  $R$  is given by (9.150), or

$$R = \sum_{e \in B} w(e) \Pr[\psi] \quad (9.97)$$

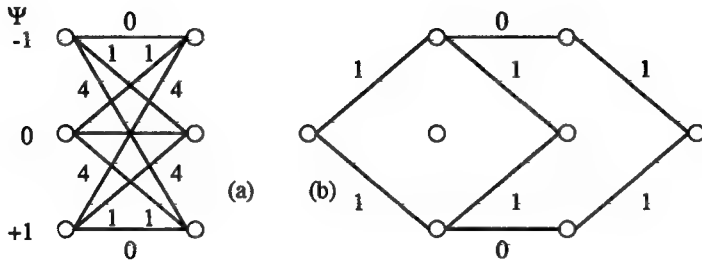
where  $B$  is the set of error events with minimum distance,  $w(e)$  is the number of symbol errors in the error event  $e$ , and  $\psi$  is the correct state trajectory.

#### Example 9-37.

Continuing Example 9-36, at any given time  $k$ , exactly one minimum-distance error event can start at that time, regardless of the correct state trajectory. If  $a_k = 0$ , which occurs with probability  $1/2$ , then the top error event is possible, corresponding to  $\epsilon_k = -1$ . If  $a_k = 1$ , which also occurs with probability  $1/2$ , then the bottom error event in Figure 9-21b is possible, corresponding to  $\epsilon_k = +1$ . Consequently,  $P = 1$ . Since the error event includes one symbol error,  $w(e) = 1$  for each  $e \in B$ , and since there is only one minimum distance error event possible for each correct path through the trellis,  $R = 1$ . Consequently,  $C = 1$  and  $\Pr[\text{symbol error}] \approx Q(\sqrt{1.25}/2\sigma)$ .  $\square$

#### Example 9-38.

Consider a channel with response  $G(z) = 1 - z^{-1}$  and a binary alphabet  $\{0, 1\}$ . The trellis is shown in Figure 9-22a. What is interesting about this channel is that two of the branches not on the correct zero path have zero branch metric. As a result, there are an infinite number of error events at the minimum distance of  $\sqrt{2}$ , corresponding to any sequence of consecutive errors of the same polarity. First note that since every correct path has at least one minimum-distance error event,  $P = 1$ . To find  $R$ , consider the contribution of an error event with  $m$  consecutive errors. This error event can only occur if  $m$  consecutive data



**Figure 9-22.** a. The trellis for the channel of Example 9-38. b. Four of the minimum-distance error events. There are actually an infinite number, corresponding to any sequence of consecutive errors with the same polarity.

symbols have the same polarity as the error, and the  $(m + 1)$  symbol has the opposite polarity. This event has probability  $2^{-(m+1)}$ . This error event contributes  $m$  symbol errors, and there are two such error events, so

$$R = 2 \times \sum_{m=1}^{\infty} m 2^{-(m+1)} = 2. \quad (9.98)$$

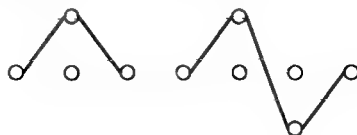
So  $C$  may be as large as 2. The behavior of this particular channel is analogous to that of *catastrophic convolutional codes*, discussed further in Chapter 13.  $\square$

In retrospect, the infinite number of minimum-distance error events of Example 9-38 is not unexpected, since a sequence of consecutive ones and a sequence of consecutive zeros both result in the same channel output sequence (all zeros). Thus, it is difficult for the ML sequence detector to distinguish the two cases; if it happens to make an error at the beginning, which is the only place where they differ, it will get the entire sequence wrong.

It may seem that an infinite number of error sequences must be considered in finding the minimum distance. In fact Figure 9-22 illustrates that minimum-distance error events can be infinite in length, although this is unusual. However, as long as  $P$  and  $R$  are moderate, it is adequate to find just the minimum distance, and necessarily not the number of error events at the minimum distance. If our goal is limited to finding the minimum distance, *there are only a finite number of error events that need be considered*. This statement follows from the following observation: if an error event passes through the same state twice (other than the all-zero path), then it need not be considered in searching for the minimum-distance error event. This is because the shorter error event obtained by removing that portion of the path between the two passes through the same state must have a path metric at least as small. This rule allows us to remove from consideration many error events. In particular, if the trellis has  $N$  states, then only error events that pass through the  $N-1$  non-zero states at most once need be considered, and such error events are of length at most  $N-1$ .

**Example 9-39.**

For a symmetric three-state trellis, corresponding to binary data symbols and  $J = 1$ , it suffices to consider the error events shown below:



By symmetry all branches that are vertical mirror images of one another have the same branch metric. There are thus two error events not shown (the mirror images) that need not be considered since they have the same path metrics. Of the two error events shown, the second will always have a path metric at least as large as the first, because of the mirror-image symmetry of the branch metrics and because of the extra branch in the middle. Thus, for a symmetric three-state trellis, the error event of length one *always* has the minimum distance. A word of caution is in order, however. From Example 9-38 we know that there can be many error events with this minimum distance.  $\square$

For a trellis with a large number of states, the number of error events that must be considered, although finite, is still large. In this case, the minimum-distance error events can be found using the Viterbi algorithm, and it is wise to use a computer.

## 9.7. SHOT NOISE SIGNAL WITH KNOWN INTENSITY

For most digital communication media the additive Gaussian noise model considered earlier in this chapter is quite appropriate. The major exception is the optical-fiber channel under some circumstances. It was pointed out in Section 5.3 that the *signal* at the output of an optical detector is actually *shot noise* or a *filtered Poisson process*. The randomness of the signal itself is therefore a contributor to errors, as is thermal noise introduced in receiver preamplifiers. The appropriate detection technique thus depends on the situation:

- When thermal noise is the dominant impairment, the white Gaussian noise detector is appropriate, since taking account of the shot noise nature of the signal will have little impact.
- When thermal noise is insignificant, as in a homodyne or heterodyne optical receiver (Chapter 8), the shot noise nature of the signal will be dominant, and the optimal detector should take it into account.

In this section we consider optimal ML detection of a shot-noise signal with time-varying known intensity, approximating the case where thermal noise is insignificant. When the shot-noise has high intensity (roughly speaking, when there are a large number of received photons per bit), we showed in Section 3.4.5 that the signal could be accurately approximated as a deterministic signal (the mean value of the shot noise) plus additive Gaussian noise. Unfortunately, the variance of the latter is time-varying (see (3.198)), and hence this noise is non-stationary and the previous results do not apply even approximately to this case. By resolving the detection problem, we

will show that the ML detector correlates against not the signal intensity as one might expect from the Gaussian case, but rather against the *logarithm* of the signal intensity. We will give a simplified derivation of this result under specific assumptions, to avoid getting overly involved in the details.

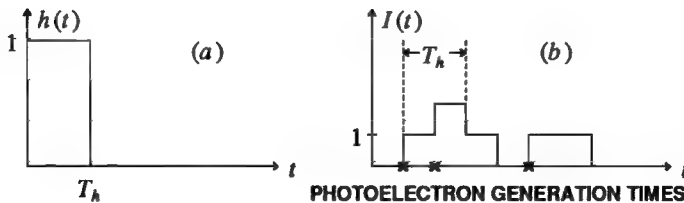
We saw in Section 5.3.4 that the current output of the photodetector is a shot noise process with intensity (average number of photoelectrons generated per unit time) proportional to the incident optical power. Assume that one of  $M$  signals is received on the interval  $[0, T]$ , where the current intensity for the  $m$ -th signal is  $\lambda_m(t) + \lambda_{\text{dark}}$  where  $\lambda_{\text{dark}}$  is the dark current that is always present, regardless of the signal. Our first simplifying assumption is that  $h(t)$ , the response of the receiver circuitry (photodetector biasing circuitry and preamplifier) to a single photoelectron assumes the particular shape of Figure 9-23a. This shape simplifies the calculation, because the photodetector output becomes a *modified counting process*, as shown in Figure 9-23b. We call it a counting process since the current  $I(t)$  at time  $t$  is equal to the count of the number of photoelectrons in the interval of time  $[t - T_h, t]$ .

For the assumed pulse shape, the current waveform is not strictly bandlimited and we cannot sample without aliasing. However, it is a reasonable approximation to sample the current every  $T_h$  seconds, yielding

$$K_n = I(nT_h), \quad (9.99)$$

which equals precisely the number of photoelectrons since the last sample; that is, the number of photoelectrons generated over the interval  $[(n-1)T_h, nT_h]$ . We choose this sampling rate for two reasons:

- The successive samples are statistically independent, since the arrival of photoelectrons follows a Poisson process and hence the number of arrivals in non-overlapping intervals are independent. If we sampled any faster, we would lose this independence, complicating the results to follow.
- This is the minimum sampling rate that avoids throwing away useful information about the signal, since with less frequent samples there would be photoelectrons that would be missed because they did not affect any sample.



**Figure 9-23.** a. The assumed shape of  $h(t)$ , the response of the detector to a single photoelectron, for calculation of the optimal receiver processing. b. An example of the sample function of the shot noise photodetector current resulting from this pulse shape.

In summary, the samples  $K_n$  are independent and Poisson distributed with parameter

$$\Gamma_{n|m} = \int_{(n-1)T_h}^{nT_h} \lambda_m(t) dt + T_h \lambda_{\text{dark}}. \quad (9.100)$$

If we assume that the response time  $T_h$  of the photodetector will be much faster than the rate of change in the intensity, as will almost always be the case, then we get the approximation

$$\Gamma_{n|m} \approx T_h \lambda_m(nT_h) + T_h \lambda_{\text{dark}}. \quad (9.101)$$

To determine the ML detector, we must calculate the probability of the received samples  $K_n$  conditioned on the  $m$ -th signal having been transmitted. The probability of one sample  $I(nT_h) = K_n$  is the Poisson distribution of (3.134),

$$p_{K_n|M}(k_n|m) = \frac{\Gamma_{n|m}^{k_n} e^{-\Gamma_{n|m}}}{k_n!}, \quad (9.102)$$

and since the samples are independent, the joint probability of  $L$  samples is the product of the probabilities,

$$p_{K_1, \dots, K_L|M}(k_1, \dots, k_L|m) = \prod_{n=1}^L \frac{\Gamma_{n|m}^{k_n} e^{-\Gamma_{n|m}}}{k_n!}, \quad (9.103)$$

where  $L$  is the number of samples required to cover the signaling interval  $[0, T]$ ,  $T = LT_h$ . For convenience, we calculate the logarithm of this probability,

$$\log_e p_{K_1, \dots, K_L}(k_1, \dots, k_L) = \sum_{n=1}^L \left[ k_n \log_e \Gamma_{n|m} - \Gamma_{n|m} - \log_e(k_n!) \right]. \quad (9.104)$$

Since the logarithm is monotonic, the ML detector chooses the signal  $m$  for which this quantity is maximum. In (9.104), the only quantity that is a function of the signal is  $\Gamma_{n|m}$ , and hence the last term can be discarded. We can make a couple of further simplifications. First, from (9.100),

$$\sum_{n=1}^L \Gamma_{n|m} = E_m + LT_h \lambda_{\text{dark}}, \quad (9.105)$$

where

$$E_m = \int_0^T \lambda_m(t) dt \quad (9.106)$$

is the integral of a quantity proportional to the incident power on the detector and hence is proportional to the total energy incident on the detector for the  $m$ -th signal. The second term in (9.105) can be ignored since it is signal independent. Also, the first term in (9.104) can be approximated by an integral, since from (9.101)

$$\begin{aligned}
 \sum_{n=1}^L k_n \log_e \Gamma_{n|m} &\approx \sum_{n=1}^L i(nT_h) \log_e (T_h (\lambda_m(nT_h) + \lambda_{\text{dark}})) \\
 &\approx \frac{1}{T_h} \int_0^T i(t) \log_e (T_h \lambda_{\text{dark}} (1 + \frac{\lambda_m(t)}{\lambda_{\text{dark}}})) dt .
 \end{aligned}
 \tag{9.107}$$

Finally, we can ignore the signal-independent  $T_h$  and  $T_h \lambda_{\text{dark}}$  terms, so that the ML detector equivalently finds the signal that satisfies

$$\max_{1 \leq m \leq M} \left\{ \Xi_m = \int_0^T I(t) \log_e \left( 1 + \frac{\lambda_m(t)}{\lambda_{\text{dark}}} \right) dt - E_m \right\} .
 \tag{9.108}$$

For high intensity, where the signal current is large relative to the dark current,

$$\Xi_m \approx \int_0^T I(t) \log_e \lambda_m(t) dt - E_m
 \tag{9.109}$$

where the constant  $\lambda_{\text{dark}}$  term has been thrown away. The ML detector correlates the photodetector current against the logarithm of signal intensity waveform. For the small signal case, where the dark current is larger than the signal, we can approximate  $\log_e(1 + \epsilon)$  by  $\epsilon$ , and thus the ML detector reduces to a correlation against the signal as in the Gaussian case,

$$\Xi_m \approx \int_0^T I(t) \lambda_m(t) dt - E_m .
 \tag{9.110}$$

#### Example 9-40.

If the signal intensity  $\lambda(t)$  is constant over an interval  $[0, T]$ , the ML detector simply integrates the current over that interval. Hence the "integrate and dump" detector of Figure 8-14 is in fact the ML detector, justifying that approach.  $\square$

## 9.8. FURTHER READING

The original Viterbi algorithm reference is [4], with a more tutorial paper following shortly thereafter [6]. A broader perspective is given in an excellent tutorial by D. Forney [7]. The Viterbi algorithm was originally derived to decode error-correcting codes (see Chapter 11). The proposal to use the Viterbi algorithm for ISI channels comes from Forney [5]. Omura [8] first pointed out that the Viterbi algorithm could be derived from the principles of dynamic programming, invented by Bellman [9]. A comprehensive coverage of its use in error-correcting codes can be found in Viterbi and Omura [10], which also has an extensive set of references.

## APPENDIX 9-A KARHUNEN-LOEVE EXPANSION

In this Appendix, we derive some of the detailed results for the Karhunen-Loeve expansion used in Section 9.3.2.

### Integral Equation

It is necessary to show that (9.37) imposes necessary and sufficient conditions for (9.35) to be satisfied. Assuming that (9.35) holds, after substituting for  $Z_i$  and  $Z_j$  from (9.36) and exchanging the order of expectation and integration,

$$\sigma_i^2 \delta_{i,j} = \int_0^T \phi_i^*(t) \int_0^T R_Z(t - \tau) \phi_j(\tau) d\tau dt . \quad (9.111)$$

A sufficient condition for this to be satisfied is (9.37), as can be established by substituting (9.37) into (9.111). To show that (9.37) is a necessary condition, we assume that (9.33) and (9.35) are valid, and show that this implies (9.37). Multiplying (9.33) by  $Z_n^*$  and taking the expected value, we get

$$E[Z(t)Z_n^*] = E\left[\sum_{i=1}^{\infty} Z_i Z_n^* \phi_i(t)\right] = \sigma_n^2 \phi_n(t) , \quad 0 \leq t \leq T . \quad (9.112)$$

Similarly, multiplying the conjugate of (9.36) by  $Z(t)$  and taking the expectation,

$$E[Z(t)Z_n^*] = E\left[Z(t) \int_0^T Z^*(\tau) \phi_n(\tau) d\tau\right] = \int_0^T R_Z(t - \tau) \phi_n(\tau) d\tau , \quad 0 \leq t \leq T . \quad (9.113)$$

Equating these two results establishes (9.37).

### Derivation of the Continuous-time Whitening Filter

We now derive (9.44). Define

$$g_l(t) = \sum_{i=1}^{\infty} \frac{s_{l,i}}{\sigma_i^2} \cdot \phi_i(t) \quad (9.114)$$

and then (9.43) follows directly by multiplying both sides of (9.114) by  $R_Z(\tau - t)$  and integrating. Similarly,

$$\int_0^T s_m(t) g_m^*(t) dt = \sum_{i=1}^{\infty} \frac{s_{m,i}^*}{\sigma_i^2} \int_0^T s_m(t) \phi_i^*(t) dt = \sum_{i=1}^{\infty} \frac{|s_{m,i}|^2}{\sigma_i^2} = E_m \quad (9.115)$$

### Sufficient Statistic Argument

In Section 9.3.3 we derived a set of sufficient statistics  $\{U_k, 1 \leq k \leq N\}$  for the received signal  $Y(t)$ ,  $0 \leq t \leq T$ , by letting  $T \rightarrow \infty$  and using intuitive arguments. Here we derive these sufficient statistics carefully for finite  $T$  using the Karhunen-Loeve

expansion. The results remain valid as  $T \rightarrow \infty$ . Define

$$f_m(t) = \sum_{i=1}^{\infty} \frac{s_{m,i}}{\sigma_i} \cdot \phi_i(t), \quad 1 \leq m \leq L, \quad 0 \leq t \leq T, \quad (9.116)$$

where  $\{\phi_i(t), \sigma_i^2, 1 \leq i < \infty\}$  are the eigenfunctions and eigenvalues of (9.37) and the  $\{s_{m,i}\}$  are given by (9.39). As  $T \rightarrow \infty$ , (9.116) approaches the same definition as (9.48). The complete orthonormal basis  $\{\psi_k(t), 1 \leq k < \infty\}$  is chosen so that the first  $N$  members are a basis for the subspace  $M_f$  spanned by the  $\{f_m(t), 1 \leq m \leq L\}$  in (9.116). Using (9.48),

$$F_{m,k} = \int_0^T f_m(t) \psi_k^*(t) dt = \sum_{i=1}^{\infty} \frac{s_{m,i}}{\sigma_i} \psi_{k,i}^* \quad (9.117)$$

where

$$\psi_{k,i} = \int_0^T \psi_k(t) \phi_i^*(t) dt. \quad (9.118)$$

Since the  $\phi_i(t)$  are a complete orthonormal set, and (9.118) are the components of  $\psi_k(t)$  in terms of the  $\phi_i(t)$ , it follows that

$$\psi_k(t) = \sum_{i=1}^{\infty} \psi_{k,i} \phi_i(t), \quad (9.119)$$

and thus

$$\begin{aligned} \delta_{k,j} &= \int_0^T \psi_k(t) \psi_j^*(t) dt = \sum_{i=1}^{\infty} \sum_{l=1}^{\infty} \psi_{k,i} \psi_{j,l}^* \int_0^T \phi_i(t) \phi_l^*(t) dt \\ &= \sum_{i=1}^{\infty} \psi_{k,i} \psi_{j,i}^*. \end{aligned} \quad (9.120)$$

Thus, the discrete-time sequences  $\{\psi_{k,i}, 1 \leq k \leq \infty\}$  for  $1 \leq i < \infty$  are orthonormal.

Returning to the received signal of (9.40), and forming the inner product of both sides with  $\psi_{k,i}, 1 \leq i \leq \infty$ , thus expressing it in terms of a new basis  $\{\psi_{k,i}, 1 \leq k \leq \infty\}$ ,

$$U_k = \sum_{i=1}^{\infty} \frac{Y_i}{\sigma_i} \psi_{k,i}^* = \sum_{i=1}^{\infty} \frac{s_{l,i}}{\sigma_i} \psi_{k,i}^* + \sum_{i=1}^{\infty} \frac{Z_i}{\sigma_i} \psi_{k,i}^*. \quad (9.121)$$

The first term on the right side of (9.121) is  $F_{l,k}$ , and the second is a noise term  $W_k$ . All that remains to establish (9.51) is to show that  $W_k$  is white, which follows from

$$E[W_k W_j^*] = E \left[ \sum_{i=1}^{\infty} \sum_{l=1}^{\infty} \frac{Z_i Z_l^*}{\sigma_i \sigma_l} \psi_{k,i}^* \psi_{j,l} \right] = \sum_{i=1}^{\infty} \psi_{j,i} \psi_{k,i}^* = \delta_{i,j}. \quad (9.122)$$

Since  $W_k$  is a linear function of a circularly symmetric process  $Z_k$ , it is circularly symmetric, and (9.122) implies that the  $W_k$  are mutually independent.



We can express the sufficient statistics in terms of continuous-time signals as follows. Substituting from (9.118) in (9.121),

$$U_k = \sum_{i=1}^{\infty} \frac{Y_i}{\sigma_i} \int_0^T \psi_k^*(t) \phi_i(t) dt = \int_0^T U(t) \psi_k^*(t) dt \quad (9.123)$$

where

$$U(t) = \sum_{i=1}^{\infty} \frac{Y_i}{\sigma_i} \cdot \phi_i(t), \quad 0 \leq t \leq T, \quad (9.124)$$

is the output of the whitening filter. This confirms (9.49).

## APPENDIX 9-B GENERAL ML AND MAP SEQUENCE DETECTORS

In Section 9.6 we illustrated the Viterbi algorithm for the additive Gaussian noise and BSC noise generation models, where the signal generator is a Markov chain. In this appendix we show that the Viterbi algorithm applies to *any* noise generator with independent noise components.

Let the random vector  $\Psi$  of length  $K+1$  denote the state sequence  $\Psi_k$  from  $k = 0$  to  $k = K$ , and let the vector  $\psi$  denote an outcome of this random vector. Similarly let the vector  $Y$  denote the observations  $Y_k$  from  $k = 0$  to  $k = K - 1$  and  $y$  an outcome (note that there is one fewer observation than states because observations correspond to transitions between states). Then given an observation  $y$ , the *MAP sequence detector* selects the  $\hat{\psi}$  that maximizes the posterior probability  $p_{\Psi|Y}(\hat{\psi} | y)$ . Note that the criterion is to maximize the *a posteriori* probability of the *whole sequence* of states, rather than a single state, and hence the term *sequence detector*.

In this appendix we will omit the subscripts in the p.d.f.'s, writing  $p(\hat{\psi} | y)$  instead of  $p_{\Psi|Y}(\hat{\psi} | y)$ , for example. There is no ambiguity here, and the shorthand will greatly simplify the expressions.

The MAP sequence detector can equivalently maximize the product  $p(\hat{\psi} | y)f(y)$  because  $f(y)$  is not dependent on our choice  $\hat{\psi}$ . (The notation  $f(y)$  implies that  $Y_k$  is continuous-valued, as in the additive Gaussian case. If it is discrete-valued, as in the BSC, then simply replace  $f(y)$  with  $p(y)$ .) From the mixed form of Bayes' rule (3.31), we can equivalently maximize  $f(y | \psi)p(\hat{\psi})$ .

### Exercise 9-3.

Show that Bayes' rule and the Markov property imply that

$$p(\hat{\psi}) = p(\hat{\psi}_0) \prod_{k=0}^{K-1} p(\hat{\psi}_{k+1} | \hat{\psi}_k). \quad (9.125)$$

□

This is intuitive because the probability of a given state trajectory  $\hat{\psi}$  is equal to the product of the probabilities of the corresponding state transitions and the probability of the initial state. Since we assume the initial state is known,  $p(\hat{\psi}_0) = 1$ . Because of the independent noise components assumption,

$$f(y|\hat{\psi}) = \prod_{k=0}^{K-1} f(y_k|\hat{\psi}_k). \quad (9.126)$$

Furthermore, since  $Y_k$  depends on only two of the states in  $\psi$ , we can write

$$f(y|\hat{\psi}) = \prod_{k=0}^{K-1} f(y_k|\hat{\psi}_{k+1}, \hat{\psi}_k). \quad (9.127)$$

Putting these results together, we wish to find the state sequence  $\hat{\psi}$  that maximizes

$$f(y|\hat{\psi})p(\hat{\psi}) = \prod_{k=0}^{K-1} p(\hat{\psi}_{k+1}|\hat{\psi}_k) \prod_{k=0}^{K-1} f(y_k|\hat{\psi}_{k+1}, \hat{\psi}_k). \quad (9.128)$$

We can equivalently maximize the logarithm,

$$\begin{aligned} \log[f(y|\hat{\psi})p(\hat{\psi})] &= \sum_{k=0}^{K-1} \log[p(\hat{\psi}_{k+1}|\hat{\psi}_k)] \\ &\quad + \sum_{k=0}^{K-1} \log[f(y_k|\hat{\psi}_{k+1}, \hat{\psi}_k)], \end{aligned} \quad (9.129)$$

or *minimize* the negative of the logarithm. To each transition  $(\hat{\psi}_k, \hat{\psi}_{k+1})$  in the trellis we assign the branch metric

$$w(\hat{\psi}_k, \hat{\psi}_{k+1}) = -\log[p(\hat{\psi}_{k+1}|\hat{\psi}_k)] - \log[f(y_k|\hat{\psi}_{k+1}, \hat{\psi}_k)]. \quad (9.130)$$

The MAP detector then calculates the path metric for each path through the trellis and finds the path with the smallest path metric.

Often the expression (9.130) for the weight of transitions in the trellis can be significantly simplified. For example, if all permissible transitions are equally likely then the first term is a constant and can be omitted. Alternatively, if we do not know the transition probabilities, we can assume the permissible transitions are equally likely and again omit this term. In either case the result is the *ML sequence detector*. In this case the branch metrics are

$$w(\hat{\psi}_k, \hat{\psi}_{k+1}) = -\log[f(y_k|\hat{\psi}_{k+1}, \hat{\psi}_k)]. \quad (9.131)$$

#### Example 9-41.

Consider a real-valued transmission with an additive Gaussian noise generator. In this case

$$\begin{aligned} f(y_k|\hat{\psi}_{k+1}, \hat{\psi}_k) &= f(y_k|s_k) \\ &= f_{N_k}(y_k - s_k) \\ &= \frac{1}{\sigma\sqrt{2\pi}} e^{-(y_k - s_k)^2/2\sigma^2}, \end{aligned} \quad (9.132)$$

where  $s_k$  is the output of the signal generator when the state transition is from  $\hat{\psi}_k$  to  $\hat{\psi}_{k+1}$ .

The branch metrics are

$$w(\hat{\psi}_k, \hat{\psi}_{k+1}) = \log(\sigma\sqrt{2\pi}) + \frac{(y_k - \hat{s}_k)^2}{2\sigma^2}. \quad (9.133)$$

The first term is independent of the decision, so equivalent branch metrics are

$$w'(\hat{\psi}_k, \hat{\psi}_{k+1}) = (y_k - \hat{s}_k)^2, \quad (9.134)$$

the squared Euclidean distance. Hence we have rederived the result from Section 9.2 in another way! Extensions to complex-valued signals are easy.  $\square$

#### Example 9-42.

Consider a BSC noise generator.

$$\begin{aligned} f(y_k | \hat{\psi}_{k+1}, \hat{\psi}_k) &= f(y_k | \hat{s}_k) \\ &= p^{d_H(y_k, \hat{s}_k)} (1-p)^{M-d_H(y_k, \hat{s}_k)} \end{aligned} \quad (9.135)$$

where  $p$  is the probability that the BSC will invert a bit, and  $M$  is the number of bits in each signal sample  $\hat{s}_k$ . The branch metrics are

$$\begin{aligned} w(\hat{\psi}_k, \hat{\psi}_{k+1}) &= -\log(p)[d_H(y_k, \hat{s}_k)] - \log(1-p)[M - d_H(y_k, \hat{s}_k)] \\ &= [\log(1-p) - \log(p)]d_H(y_k, \hat{s}_k) - M\log(1-p). \end{aligned} \quad (9.136)$$

The last term is not a function of the decision, and so can be ignored. Assuming  $p < 1-p$ , an equivalent branch metric is the Hamming distance

$$w'(\hat{\psi}_k, \hat{\psi}_{k+1}) = d_H(y_k, \hat{s}_k). \quad (9.137)$$

Hence we have again rederived the result from Section 9.2 in another way!  $\square$

## APPENDIX 9-C

### BIT ERROR PROBABILITY FOR SEQUENCE DETECTORS

We showed in Section 9.6 that the probability of sequence error is easy to obtain using the vector channel results from Section 7.2.4, but is often useless because the probability approaches unity as the sequence gets large. Instead of the probability of sequence error, we can compute the probability that an *error event* begins at a particular time. This effectively normalizes the probability of sequence error per unit time. Error events are defined in Section 9.6.4.

We are most often interested however in the probability of a *bit* or *symbol* error rather than an error event. In this appendix we derive the error event probability and a general expression for the probability of bit or symbol error that does not depend on linearity in the system. We then show that for the additive Gaussian white noise case the probability of error is approximately  $C \cdot Q(d_{\min}/2\sigma)$ , where  $C$  is a constant that we can easily bound, and  $d_{\min}$  is the distance of the minimum distance error event. For the BSC channel case, the probability of error is approximately  $C \cdot Q(d_{\min}, p)$ , where  $Q(\cdot, \cdot)$  is defined by (9.25).

After the sequence detector selects a path through the trellis, the receiver must translate this path into its corresponding bit sequence (recall the one-to-one mapping between incoming bit sequences and state trajectories). Several bit or symbol errors may occur as a consequence of each error event. Let  $E$  denote the set of all error events starting at time  $i$ . Each element  $e$  of  $E$  is characterized by both a correct path  $\psi$  and an incorrect path  $\hat{\psi}$  that diverge and remerge some time later. We make a stationarity assumption that  $\Pr[e]$  is independent of  $i$ , the starting time of the error event. This will of course not be true if the trellis is finite, but if it is long relative to the length of the significant error events then the approximation is accurate. Each error event causes one or more *detection errors*, where a detection error at time  $k$  means that  $X_k$  at stage  $k$  of the trellis is incorrect. For the ISI example, each  $X_k$  is a symbol  $A_k$ , so a detection error is the same as a symbol error. For the binary coding examples, each  $X_k$  is a set of one or more bits. Define

$$c_m(e) = \begin{cases} 1; & \text{if } e \text{ has a detection error in position } m \text{ (from the start } i) \\ 0; & \text{otherwise} \end{cases} \quad (9.138)$$

This function characterizes the sample times corresponding to detection errors in error event  $e$ .

#### Example 9-43.

Consider Example 9-25, the ISI example. Let  $e_1$  denote the error event of Figure 9-19a, which assumes that the correct state trajectory  $\psi$  consists of zero states. From Figure 9-17b we see that this error event causes decisions  $\hat{x}_i = 1$  and  $\hat{x}_{i+1} = 0$ . Since  $x_i = 0$  and  $x_{i+1} = 0$  are the correct decisions,

$$c_m(e_1) = \delta_m. \quad (9.139)$$

□

The probability of a particular error event  $e$  starting at time  $i$  and causing a detection error at time  $k$  is

$$c_{k-i}(e) \Pr[e]. \quad (9.140)$$

Since the error events in  $E$  are disjoint (if one occurs no other can occur),

$$\Pr[\text{detection error at time } k] = \sum_{i=-\infty}^k \sum_{e \in E} \Pr[e] c_{k-i}(e). \quad (9.141)$$

Exchanging the order of summation, assuming this is legitimate,

$$\Pr[\text{detection error at time } k] = \sum_{e \in E} \Pr[e] \sum_{i=-\infty}^k c_{k-i}(e). \quad (9.142)$$

By a change of variables,

$$w(e) = \sum_{i=-\infty}^k c_{k-i}(e) = \sum_{m=0}^{\infty} c_m(e) \quad (9.143)$$

which is the total number of detection errors in  $e$ . Thus

$$\Pr[\text{detection error}] = \sum_{e \in E} \Pr[e] w(e), \quad (9.144)$$

where we note that the dependence on  $k$  has disappeared. Hence, the probability of a detection error at any particular time is equal to the expected number of detection errors caused by error events starting at any fixed time  $i$ . In retrospect this result is not unexpected, since from the perspective of time  $k$ , the probability of a detection error at that time must take into account all error events starting at times prior to  $k$ .

The probability of the error event  $e$  depends on the probabilities of both the correct and incorrect paths  $\psi$  and  $\hat{\psi}$  that make up  $e$ ,

$$\Pr[e] = \Pr[\psi] \Pr[\hat{\psi} | \psi]. \quad (9.145)$$

It is usually difficult to find exact expressions for  $\Pr[\hat{\psi} | \psi]$ , but bounds are often easy. The reason is easy to see in the simple example of Figure 9-24, where we assume there are only three possible trajectories. In Figure 9-24a the ML decision regions for the three signals are shown. These decision regions lie in a  $K$ -dimensional space that we schematically represent on the two-dimensional page. Now suppose that  $\psi$  is the actual trajectory, corresponding to signal  $s$  in Figure 9-24a. The region corresponding to the detection of  $\hat{\psi}$  is shown in Figure 9-24b. The probability of the noise carrying us into this region is very difficult to calculate, especially as the number of possible trajectories gets large. However, this probability is easy to upper bound by using the larger decision region of Figure 9-24c, which ignores the possibility of any trajectory other than  $\psi$  and  $\hat{\psi}$ .

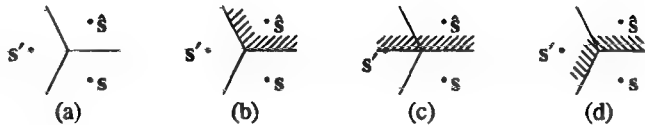
For the additive white Gaussian noise model, the probability of the region in Figure 9-24c is

$$\Pr[\hat{\psi} | \psi] \leq Q(d(\hat{\psi}, \psi)/2\sigma) \quad (9.146)$$

where  $d(\hat{\psi}, \psi)$  is the Euclidean distance between transmitted signals  $\hat{s}$  and  $s$  corresponding to state trajectories  $\hat{\psi}$  and  $\psi$ . For the BSC,

$$\Pr[\hat{\psi} | \psi] \leq Q(d(\hat{\psi}, \psi), p) \quad (9.147)$$

where  $d(\hat{\psi}, \psi)$  is now a Hamming distance and  $Q(\cdot, \cdot)$  is defined by (9.25). The bound is precisely the probability that the received signal is closer to the signal  $\hat{s}$



**Figure 9-24.** a. Three signals corresponding to state trajectories  $\psi$ ,  $\hat{\psi}$  and  $\psi'$  where  $\psi$  is the actual state trajectory. b. If received signal (with noise) is in the shaded region, the ML detector will choose trajectory  $\hat{\psi}$ . c. The decision region for  $\hat{\psi}$  if there were only two signals,  $\psi$  and  $\hat{\psi}$ . d. If the received signal (with noise) is in the shaded region, the ML detector will generate an error event.

corresponding to  $\hat{\psi}$  than it is to the signal  $s$  corresponding to the correct path  $\psi$ . Combining (9.144), (9.145), and (9.146), for the Gaussian case

$$\Pr[\text{detection error}] \leq \sum_{e \in E} w(e) \Pr[\psi] Q(d(\hat{\psi}, \psi)/2\sigma). \quad (9.148)$$

This can be written

$$\Pr[\text{detection error}] \leq \sum_{e \in B} w(e) \Pr[\psi] Q(d_{\min}/2\sigma) + \text{other terms} \quad (9.149)$$

where  $B$  is the subset of error events in  $E$  that have distance  $d_{\min}$ , and the "other terms" all have arguments to the  $Q(\cdot)$  function larger than  $d_{\min}/2\sigma$ . At high SNR these other terms become insignificant and the upper bound on  $\Pr[\text{detection error}]$  approaches  $RQ(d_{\min}/2\sigma)$ , where

$$R = \sum_{e \in B} w(e) \Pr[\psi]. \quad (9.150)$$

For the BSC case, just replace  $Q(d_{\min}/2\sigma)$  with  $Q(d_{\min}, p)$ .

#### Example 9-44.

In Example 9-43 we considered the error event  $e_1$  shown in Figure 9-19a, and found that  $w(e_1) = 1$  and the distance is  $\sqrt{1.25}$ . This is the minimum distance, and it occurs for the eight error events shown in Figure 9-25, each of which also have  $w(e) = 1$ . Consequently, (9.150) becomes

$$R = \sum_{e \in B} \Pr[\psi] \quad (9.151)$$

where  $B$  is the set of error events shown in Figure 9-25. Assuming all possible actual paths are equally likely, and noting that only three successive states of each  $\psi$  in Figure 9-25 are specified,  $\Pr[\psi] = 2^{-3} = 1/8$  for each one. Hence  $R = 1$  and (9.149) becomes

$$\Pr[\text{detection error}] \leq Q(\sqrt{1.25}/2\sigma) + \text{other terms}. \quad (9.152)$$

We can get an idea of the magnitude of the "other terms" by considering the set of second most probable error events, one of which is shown in Figure 9-19b. It has length two, distance  $d = \sqrt{3.5}$  from the correct path, weight  $w(e_2) = 2$ , and  $\Pr[\psi] = 1/16$ , so its contribution to the sum is  $0.125Q(\sqrt{3.5}/2\sigma)$ . Because of the exponential decrease in  $Q(\cdot)$ , this term is orders of magnitude smaller than the first for small  $\sigma$ . Consequently, we conclude that the "other terms" in (9.152) can safely be ignored.  $\square$

A lower bound on the probability of detection error for the additive white Gaussian noise model can also be found. Combining this with the upper bound in (9.148) leads

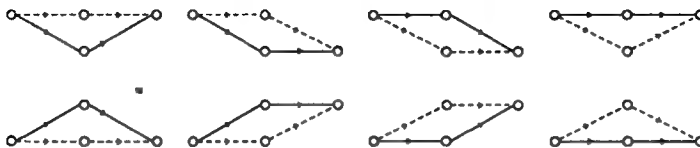


Figure 9-25. Eight error events that have the same probabilities.

to an accurate estimate of  $\Pr[\text{detection error}]$ . Returning to Figure 9-24, we want to use the decision region of Figure 9-24c to somehow obtain a lower bound. Shown in Figure 9-24d is the decision region corresponding to *any error event* conditioned on actual state sequence  $\psi$ . The probability of any error event is evidently lower bounded by calculating the probability of the smaller decision region in Figure 9-24c. Thus, we see that in order to determine a lower bound, we must start with the probability of *any* error event, rather than the probability of a particular error event.

Since  $w(e) \geq 1$  for all error events  $e$ , then from (9.144)

$$\Pr[\text{detection error}] \geq \sum_{e \in E} \Pr[e] = \Pr[\text{an error event}]. \quad (9.153)$$

Now consider a particular actual path  $\psi$  through the trellis. For this path, let  $d_{\min}(\psi)$  denote the distance of the minimum distance error event (either Euclidean or Hamming). Of course,  $d_{\min}(\psi) \geq d_{\min}$ , where  $d_{\min}$  is the minimum distance error event over all possible actual state sequences  $\psi$ . As in Figure 9-24c, if  $\psi$  is the actual state sequence, the probability of an error event is lower bounded by Figure 9-24c. Obviously to make this bound strongest, we want to choose a particular error event that is closest to  $\psi$ , one of those at distance  $d_{\min}(\psi)$ . Hence, for the Gaussian case

$$\Pr[\text{an error event} | \psi] \geq Q(d_{\min}(\psi)/2\sigma). \quad (9.154)$$

Combining this with (9.153) we get

$$\Pr[\text{detection error} | \psi] \geq Q(d_{\min}(\psi)/2\sigma). \quad (9.155)$$

Consequently,

$$\Pr[\text{detection error}] \geq \sum_{\psi} \Pr[\psi] Q(d_{\min}(\psi)/2\sigma). \quad (9.156)$$

If we omit some terms in this summation, the bound will still be valid since the terms are all non-negative. Thus, let us retain only those state sequences  $\psi$  for which  $d_{\min}(\psi) = d_{\min}$ ,

$$\Pr[\text{detection error}] \geq \sum_{\psi \in A} \Pr[\psi] Q(d_{\min}/2\sigma), \quad (9.157)$$

where  $A$  is the set of actual paths  $\psi$  that have a minimum distance error event, and  $d_{\min}$  is that minimum distance. Define

$$P = \sum_{\psi \in A} \Pr[\psi], \quad (9.158)$$

the probability that a randomly chosen  $\psi$  has an error event starting at any fixed time with distance  $d_{\min}$  (or is consistent with a minimum distance error event). Then

$$\Pr[\text{detection error}] \geq PQ(d_{\min}/2\sigma). \quad (9.159)$$

In retrospect this lower bound is intuitive, since we would expect that every state sequence consistent with a minimum-distance error event will result in a probability of error event at least as large as  $Q(d_{\min}/2\sigma)$ , and each error event will result in at least one detection error. For the common case where all possible paths  $\psi$  through the trellis are consistent with a minimum-distance error event,  $P = 1$ . This is true in

## Example 9-44.

Combining our upper and lower bounds,

$$PQ(d_{\min}/2\sigma) \leq \Pr[\text{detection error}] \leq RQ(d_{\min}/2\sigma), \quad (9.160)$$

where the upper bound is approximate since some terms were thrown away. We conclude that at high SNR

$$\Pr[\text{detection error}] \approx C \cdot Q(d_{\min}/2\sigma), \quad (9.161)$$

for some constant  $C$  between  $P$  and  $R$ . The BSC case is identical,

$$\Pr[\text{detection error}] \approx C \cdot Q(d_{\min}, p) \quad (9.162)$$

where  $Q(\cdot, \cdot)$  is defined in (9.25).

## Example 9-45.

Continuing Example 9-44, note that  $P = R = 1$ . Hence

$$\Pr[\text{detection error}] \approx Q(\sqrt{1.25}/2\sigma). \quad (9.163)$$

For this example, each detection error causes exactly one bit error, so  $\Pr[\text{detection error}] = \Pr[\text{bit error}]$ . Hence, with the sequence detector we get approximately the same probability of error as for an isolated pulse and a matched filter receiver (see Problem 9-9).

□

In general, a single detection error may cause more than one bit error. Suppose each input to the Markov chain  $X_k$  is determined by  $n$  source bits (and hence comes from an alphabet of size  $2^n$ ). Then each detection error causes at least one and at most  $n$  bit errors. Hence we can write

$$\frac{1}{n} \Pr[\text{detection error}] \leq \Pr[\text{bit error}] \leq \Pr[\text{detection error}]. \quad (9.164)$$

Typically we make the pessimistic assumption that

$$\Pr[\text{bit error}] \approx \Pr[\text{detection error}]. \quad (9.165)$$

## PROBLEMS

- 9-1. Suppose a binary symbol  $A$  ( $\Omega_A = \{0,1\}$ ) with  $p_A(0) = q$  and  $p_A(1) = 1 - q$  is transmitted through the BSC of Figure 9-2. The observation  $Y$  is also a binary symbol ( $\Omega_Y = \{0,1\}$ ); it equals  $A$  with probability  $1 - p$ .
- Find the ML detection rule. Assume  $p < 1/2$ .
  - Find the probability of error of the ML detector as a function of  $p$  and  $q$ .
  - Assume  $p = 0.2$  and  $q = 0.9$ . Find the MAP detector and its probability of error. Compare this probability of error to that in part (b).
  - Find the general MAP detector for arbitrary  $p$  and  $q$ .
  - Find the conditions on  $p$  and  $q$  such that the MAP detector always selects  $\hat{d} = 0$ .



- 9-2. Consider the vector detection problem for the BSC of Example 9-15. Specify the MAP detector for some given prior probabilities for signal vectors.
- 9-3. Assume the random variable  $X$  has sample space  $\Omega_X = \{-3, -1, +1, +3\}$  with prior probabilities  $p_X(\pm 3) = 0.1$  and  $p_X(\pm 1) = 0.4$ . Given an observation  $y$  of the random variable  $Y = X + N$ , where  $N$  is a zero mean Gaussian random variable with variance  $\sigma^2$ , independent of  $X$ , find the decision regions for a MAP detector. Now suppose  $\sigma^2 = 0.25$  and  $y = 2.1$ . What is the decision?
- 9-4. Assume the random variable  $X$  is from the alphabet  $\Omega_X = \{x_1, x_2\}$ . Define the random variable  $Y = X + N$ , where  $N$  is a zero mean Gaussian random variable with variance  $\sigma^2$ , independent of  $X$ . Give an expression for the MAP decision boundary between  $x_1$  and  $x_2$ .
- 9-5. Consider  $M$  vectors each a distance  $d_{\min}$  from the other vectors. Assume an ML detector will be used to distinguish between these vectors.
- Give an example for  $M = 3$  of such a set of vectors where  $d_{\min}$  is a Euclidean distance of  $\sqrt{2}$ .
  - Give an example for  $M = 3$  of such a set of vectors where  $d_{\min}$  is a Hamming distance of 2.
  - Use the union bound to find an upper bound on the probability of error for your two examples, assuming additive white Gaussian noise for (a) and a BSC for (b). First give the bound assuming  $s_1$  is transmitted, then give the bound without this assumption.
- 9-6. Suppose you are given  $N$  observations  $x_1, \dots, x_N$  of the zero mean independent Gaussian random variables  $X_1, \dots, X_N$ . Assume that the random variables have the same (unknown) variance  $\sigma^2$ . What is the ML estimator for the variance?
- 9-7. Given a Gaussian channel with independent noise components, one of the following four equally likely signals is transmitted:  $(1, 1), (1, -1), (-1, 1), (-1, -1)$ . Determine the *exact* probability of error of an ML detector for Gaussian noise with variance  $\sigma^2$ .
- 9-8.
- Repeat Problem 9-7 for a BSC with error probability  $p$  and four equally likely signals:  $(000000), (000111), (111000), (111111)$ .
  - What is this error probability when  $p = 0.1$ ? Compare to the minimum distance approximation.
- 9-9. Suppose that a symbol  $A$  from the alphabet  $\Omega_A = \{0, 1\}$  is transmitted through the LTI system with impulse response

$$h_k = \delta_k + 0.5\delta_{k-1} \quad (9.166)$$

and corrupted by additive white Gaussian noise with variance  $\sigma^2$ .

- Determine the structure of the ML detector.
  - Calculate the probability of error for the ML detector.
- 9-10. Consider the system in Figure 9-26. Assume  $Z_k$  is a sequence of independent zero mean Gaussian random variables with variance  $\sigma^2$ . Assume the symbol alphabet is  $\Omega_A = \{0, 1\}$  and that the channel impulse response is

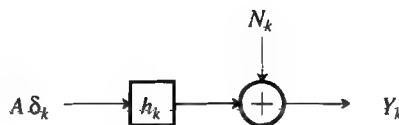


Figure 9-26. A simple case of a discrete-time channel with intersymbol interference.

$$h_k = \delta_k - 0.5\delta_{k-1} + 0.1\delta_{k-2}. \quad (9.167)$$

- (a) Derive the matched filter receiver.
- (b) Give an expression for the probability of error.
- (c) Now suppose  $\Omega_A = \{-1, 0, +1\}$ . Repeat part (a).

**9-11.**

- (a) Determine the optimal incoherent receiver structure for passband PAM modulation, where the data symbols assume only two values  $A_0 = 0$  and  $A_0 = 1$ . This is known as amplitude shift keying (ASK).
- (b) Discuss the conditions under which passband PAM can be successfully demodulated using an incoherent receiver.
- (c) Find an expression for the probability of error of the type derived in Example 9-22. You do not need to evaluate this expression.

**9-12.**

- (a) Derive a discrete-time channel model analogous to Figure 9-13 where instead of a matched filter, a general filter  $F^*(j\omega)$  is used, the Gaussian noise added on the channel has power spectrum  $N_0 S_N(\omega)$ , and symbol-rate sampling is used at the output of the filter.
- (b) Determine a model for the special case where the matched filter optimized for the particular noise spectrum is used.
- (c) As a check, verify that your model reduces to Figure 9-13 when the filter is a matched filter and the noise is white.

**9-13.** Suppose that for a channel with additive white noise, the response at the output of the whitened matched filter is  $G(z)$ . Find an example of a received pulse  $h(t)$  that results in this response.

**9-14.** Find the appropriate branch metric for the ML sequence detector for the following signal and noise generators. Let the signal generator output be non-negative,  $S_k \geq 0$  for all  $k$ . Assume the noise generator outputs conditioned on a particular signal are independent and identically distributed Poisson random variables, and the  $k$ -th output has mean value  $\alpha S_k$ . (This model is roughly equivalent to what might be encountered on a fiber optics channel.)

**9-15.** Repeat Problem 9-14 for the following. The signal generator output is binary, assuming the values  $\{0, 1\}$ , and the channel is the binary-erasure channel of Problem 4-8, with independent channel uses.

**9-16.** Consider transmitting bits  $X_k$  (zeros and ones) over a channel with additive white Gaussian noise. Assume that  $X_k = 0$  for  $k < 0$  and  $k \geq K$ . Suppose  $K = 3$  and the observation sequence is  $y_0 = 0.6$ ,  $y_1 = 0.9$ ,  $y_2 = 1.3$ , and  $y_3 = 0.3$ .

- (a) Find the ML decision sequence  $\hat{x}_k$  assuming that the additive noise is the only degradation (no ISI) and that  $X_k$  are i.i.d.
- (b) Suppose you are told that the ISI channel of Example 9-25 is being used. Draw the trellis for the Markov model and label the transition weights. What is the ML detection of the incoming bit sequence?

**9-17.** Consider Example 9-25, where  $X_k$  is an i.i.d. sequence with each sample equally likely to be zero or one. Assume an ML sequence detector, and assume that the only error event likely to occur is the one in Figure 9-19a.

- (a) Compare the probability of this error event to the probability of error in a similar binary system with no ISI, Figure 9-15 with  $g_k$  replaced with

$$g'_k = \delta_k. \quad (9.168)$$

Also assume an ML detector for this system. **Hint:** Find the difference in the noise variance  $\sigma^2$  so that the two systems yield the same probability of error. Express the answer in dB.

- (b) Find the power of  $S_k$  in Example 9-25 and in the system with no ISI in part a. Find  $K$  such that if

$$h'_k = K \delta_k \quad (9.169)$$

then both systems produce  $S_k$  with the same power.

- (c) Assume

$$h'_k = K \delta_k \quad (9.170)$$

where  $K$  is derived in part (b). Compare the probability of the error event in Example 9-25 with the probability of error in this system.

- 9-18. Consider the system in Figure 9-15 with

$$g_k = \delta_k - 0.5\delta_{k-1} + 0.1\delta_{k-2}. \quad (9.171)$$

Assume  $A_k$  is equally likely to be 0 or 1, and the  $A_k$  are independent for all  $k$ . Assume additive Gaussian white noise with variance  $\sigma^2$ .

- Model the system as a shift register process and draw the state transition diagram. Label the arcs with the input/output pair  $(A_k, S_k)$ .
- Draw one stage of a trellis and label with the input/output pairs  $(A_k, S_k)$ .
- Assume  $\psi_k = 0$  for  $k \leq 0$  and  $k \geq 5$ . Suppose the observation sequence is  $y_0 = 0.5$ ,  $y_1 = -0.2$ ,  $y_2 = 0.9$ ,  $y_3 = 1.2$ , and  $y_4 = 0.1$ . Draw a complete trellis with branch weights labeled.
- Use the Viterbi algorithm to find the ML decision sequence.
- Assuming that the correct state trajectory is  $\psi_k = 0$  for all  $k$ , find the minimum-distance error event and its distance.
- Argue convincingly that the minimum distance found in part (e) is the minimum distance for any correct state trajectory.

- 9-19. Consider a response  $G(z) = 1 + g_1z^{-1} + g_2z^{-2}$  at the output of a whitened matched filter. Assume the data symbols have alphabet  $\{0,1\}$ .

- Draw the trellis diagram that can be used to find the minimum distance, and label each transition with the appropriate branch metric.
- Specify a finite set of error events that it suffices to consider in searching for the minimum distance.
- Give an example of a channel such that the minimum-distance error event is of length less than  $(1 + g_1^2 + g_2^2)$ .

- 9-20. For a response  $G(z) = 1 + dz^{-1} + dz^{-2}$  at the whitened matched filter output, find the minimum distance as a function of  $0 \leq d \leq 1$  assuming binary signaling from alphabet  $\{0,1\}$ .

- 9-21. For the response  $G(z) = 1 + z^{-1}$ , find the set of minimum-distance error events and the resulting bound on the probability of symbol error assuming binary signaling with alphabet  $\{0,1\}$ . Can you give an intuitive explanation for the error events you find?

- 9-22.

- Apply the Chernov bound to the problem of detecting a binary shot noise signal with two known intensity functions as well as dark current. In particular, assume that the shot noise signal has filtering function  $f(t)$ , which is a composite of the photodetector response and any post-filtering, and that this signal is directly sampled at  $t = 0$  and applied to a slicer.
- Minimize the upper bound with respect to  $f(t)$ , and show that the resulting detector correlates the logarithm of the known intensity against the delta-function shot noise signal.
- Evaluate the upper bound for the solution of (b)

- 9-23. Using the results of Appendix 9-A, develop the following relationship between  $\{V_l, 1 \leq l \leq L\}$  and  $\{U_k, 1 \leq k \leq N\}$ , which serves to establish that  $\{V_l, 1 \leq l \leq L\}$  is also a set of sufficient statistics,

$$V_l = \sum_{k=1}^N F_{l,k}^* U_k, \quad 1 \leq l \leq L. \quad (9.172)$$

## REFERENCES

1. H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, Wiley, New York (1968).
2. J. G. Proakis, *Digital Communications, Second Edition*, McGraw-Hill Book Co., New York (1989).
3. S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1987).
4. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Information Theory* IT-13 pp. 260-269 (April 1967).
5. G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. on Information Theory* IT-18 pp. 363-378 (May 1972).
6. A. J. Viterbi, "Convolutional Codes and their Performance in Communication Systems," *IEEE Trans. on Communication Tech.* COM-19 pp. 751-772 (Oct. 1971).
7. G. D. Forney, Jr., "The Viterbi Algorithm," *Proceedings of the IEEE* 61(3) (March 1973).
8. J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE Trans. on Information Theory* IT-15 pp. 177-179 (1969).
9. R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ (1957).
10. A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill (1979).

# 10

---

## EQUALIZATION

---

In PAM, intersymbol interference (ISI) results from linear amplitude and phase distortion in the channel that broadens the pulses and causes them to interfere with one another. The Nyquist criterion specifies a frequency-domain condition on the received pulses under which there is no intersymbol interference. Generally this or a similar condition is not satisfied unless we *equalize* the channel, meaning roughly that we filter to compensate for the channel distortion. Unfortunately, any equalization of amplitude distortion also enhances or amplifies any noise introduced by the channel, called *noise enhancement*. There is therefore a tradeoff between accurately minimizing intersymbol interference and minimizing the noise at the slicer input. Ultimately, of course, our goal is to minimize the probability of error.

In Chapters 7, 8, and 9, we explained the maximum-likelihood sequence detector (MLSD), which is the optimal receiver for detecting a sequence of data symbols under the maximum-likelihood criterion. The MLSD does not equalize the ISI, but rather filters to ensure that the ISI is causal and monic; the resulting front-end structure is identical to the whitened matched filter (WMF) described in Chapter 8. The detection is performed by comparing the WMF output sequence against what that output sequence would be for each feasible sequence of input data symbols in the absence of noise; it chooses the data symbols that best match the WMF output according to a Euclidean distance measure.

The MLSD is not widely used for detecting PAM because there are simpler receiver structures based on equalization that perform well, and also because it has

been shown that in combination with channel coding (Chapter 14) it is not even necessary to use the MLSD to approach the best performance. In this chapter we describe several basic approaches to equalization: linear equalization, decision-feedback equalization, and transmitter precoding. In Section 10.1, the optimal equalization structures are derived under a *zero-forcing (ZF) criterion*, which means that we completely eliminate the ISI at the slicer input. The starting point is the WMF, which is used as the front-end of the receiver of Chapter 9, and relies on the sufficient statistic argument. In Section 10.2 these results are generalized by relaxing the assumption that the WMF is used, and also by considering an alternative *mean-square error (MSE) criterion*. The MSE criterion reduces noise enhancement by allowing residual ISI at the slicer input, and attempts to minimize the sum of the ISI and noise. In Section 10.3 we demonstrate that the equalization, including the matched filter in the WMF, can be implemented in discrete time if the sampling rate is increased to some multiple of the symbol rate, as is often done in practice. Section 10.4 describes briefly a practical equalizer filter structure called the *transversal filter*, which is just another name for a direct-form FIR filter. The transversal filter plays a central role in the realization of *adaptive equalization* as described in Chapter 11. Finally, in Section 10.5 we consider the channel capacity of a channel with ISI, and derive *Price's result*, which suggests that the DFE receiver structure does not compromise channel capacity at high SNR.

In this chapter we assume that the linear distortion of the channel is exactly known. Furthermore, we do not constrain the complexity of the receiver or its filter structures. Both of these assumptions are relaxed in Chapter 11, where we describe constrained-complexity filters, known as *adaptive equalizers*, that adapt to unknown or changing channel conditions.

## Notation

Some of the expressions in this chapter become fairly complicated, so will often use a simplified notation in which the frequency variable is suppressed. For a transfer function  $F(z)$ , which becomes  $F(e^{j\omega T})$  when evaluated on the unit circle, we will write simply  $F$ . Similarly, the reflected transfer function  $F^*(1/z^*)$ , which becomes  $F^*(e^{j\omega T})$  when evaluated on the unit circle, will be written in shorthand notation as  $F^*$ .

### Example 10-1.

Given a transfer function  $S(z)$  that is non-negative real on the unit circle, the minimum-phase spectral factorization can be written in four ways,

$$S(z) = A_s^2 G_s(z) G_s^*(1/z^*), \quad S(e^{j\omega T}) = A_s^2 |G_s(e^{j\omega T})|^2, \quad (10.1)$$

$$S = A_s^2 G_s G_s^*, \quad S = A_s^2 |G_s|^2. \quad (10.2)$$

We will often use in this chapter the shorthand notations of (10.2).  $\square$

Also, in this chapter the arithmetic and geometric means of a function will arise frequently, and we will use a shorthand notation for these means. Given a *non-negative real-valued* function  $f(x)$  and a subset  $X$  of the  $x$  axis, define  $|X|$  as the total size (measure) of the set  $X$ . For example, if  $X = \{x : |x| \leq x_0\}$ , then

$|X| = 2 \cdot x_0$ . The *arithmetic mean* of  $f(x)$  over  $X$  is defined as

$$\langle f \rangle_{A,X} = \frac{1}{|X|} \int_X f(x) dx, \quad (10.3)$$

which has the interpretation as the average value of  $f(x)$  over the interval  $X$ . Similarly, define a *geometric mean* of  $f(x)$  over  $X$

$$\langle f \rangle_{G,X} = \exp \left\{ \frac{1}{|X|} \int_X \log_e f(x) dx \right\}. \quad (10.4)$$

Actually the geometric mean is unchanged if we change the base of both the exponential and the logarithm; this is clear if we write (10.4) in the form

$$\log \langle f \rangle_{G,X} = \frac{1}{|X|} \int_X \log f(x) dx, \quad (10.5)$$

which is valid for any logarithm base since the logarithm appears on both sides. Thus, there is no need to specify the base of the logarithm in the geometric mean.

The integral inequality

$$\int_X \log f(x) dx \leq \log \int_X f(x) dx \quad (10.6)$$

implies that

$$\langle f \rangle_{G,X} \leq \langle f \rangle_{A,X} \quad (10.7)$$

with equality if and only if  $f(x)$  is a constant over the set  $X$ .

#### Example 10-2.

The constant  $A_s^2$  in the geometric mean formula of (10.2) is, from (2.57),

$$A_s^2 = \langle S \rangle_{G,(-\pi T, \pi T)}, \quad (10.8)$$

where the independent variable in this case is  $\omega$  rather than  $x$ .  $\square$

The arithmetic and geometric means have several useful properties. For both types of mean, the mean of a real-valued constant is itself,

$$\langle a \rangle_{A,X} = a. \quad (10.9)$$

Similarly,

$$\langle a \cdot f \rangle_{A,X} = a \cdot \langle f \rangle_{A,X}, \quad \langle a \cdot f \rangle_{G,X} = a \cdot \langle f \rangle_{G,X}. \quad (10.10)$$

The arithmetic mean also obeys the distributive law,

$$\langle a \cdot f + b \cdot g \rangle_{A,X} = a \cdot \langle f \rangle_{A,X} + b \cdot \langle g \rangle_{A,X}, \quad (10.11)$$

but the geometric mean does not. Conversely, the geometric mean obeys the multiplicative laws

$$\langle f \cdot g \rangle_{G,X} = \langle f \rangle_{G,X} \cdot \langle g \rangle_{G,X}, \quad \langle f/g \rangle_{G,X} = \frac{\langle f \rangle_{G,X}}{\langle g \rangle_{G,X}}, \quad (10.12)$$

which the arithmetic mean does not.

## 10.1. OPTIMAL ZERO-FORCING EQUALIZATION

In Chapter 9 we derived receivers for PAM with ISI based on optimal detection criteria. In particular, the MLSD offers a computational load that is constant with time. However, in practice the MLSD is rarely used to equalize ISI for a couple of reasons. First, the equalizer structures covered in this section, although offering a noise performance in the absence of channel coding that is inferior to the MLSD, are considerably less complex to implement. Second, in high performance data communication systems, error-prevention coding is almost always used (Chapters 13 and 14), and surprisingly it has been shown that coding can be used to approach channel capacity in the presence of ISI without the need to use the MLSD for equalization.

In this section, we will first review the results of Chapter 9 as they apply to PAM with ISI, and in the process develop some useful bounds on the performance of receivers in the presence of ISI. Following this background, we will describe three practical and widely used equalization techniques — linear equalization, the decision-feedback equalization, and transmitter precoding — and compare their performance to one another and to the MLSD.

### 10.1.1. Background Results

The results of Chapter 9 will prove very useful as a starting point for the derivation of optimal equalizer structures for ISI. Not only does Chapter 9 establish a canonical front end for all receivers, including those based on equalization, but it also establishes some useful bounds on the performance of such receivers.

#### Whitened Matched Filter

In Section 9.4 we established that the whitened matched filter (WMF), first encountered in Section 7.3 as an embodiment of the minimum-distance receiver design criterion, develops a set of sufficient statistics for the received signal. When the noise on the channel is white and Gaussian, the WMF can be used as the front end of a receiver designed according to any criterion of optimality. This result is particularly valuable because of the desirable properties of the WMF output:

- It is a sampled data signal, with sampling rate equal to the symbol rate.
- The equivalent discrete-time channel model is causal and minimum-phase, the latter implying that the energy of the impulse response is maximally concentrated in the early samples (Problem 2-23).
- The equivalent additive noise is Gaussian, circularly symmetric and white, implying that the samples of this complex-valued noise are independent.



These properties greatly simplify the derivation of the remainder of the receiver for any particular criterion.

For convenience, the WMF receiver front end and its equivalent channel model are repeated in Figure 10-1. The case we have shown corresponds to zero-mean white Gaussian noise on the channel with power spectrum  $N_0$ , in which case the variance of the complex-baseband noise  $Z_k$  is  $2\sigma^2$  where  $\sigma^2 = N_0/A_h^2$ . These results are based on the minimum-phase spectral factorization of the folded spectrum  $S_h(z)$ ,

$$S_h(z) = A_h^2 G_h(z) G_h^*(1/z^*), \quad (10.13)$$

where  $G_h(z)$  is a causal monic ( $G_h(z = \infty) = 1$ ) loosely minimum-phase transfer function and the positive constant  $A_h^2$  can be determined from the geometric mean formula

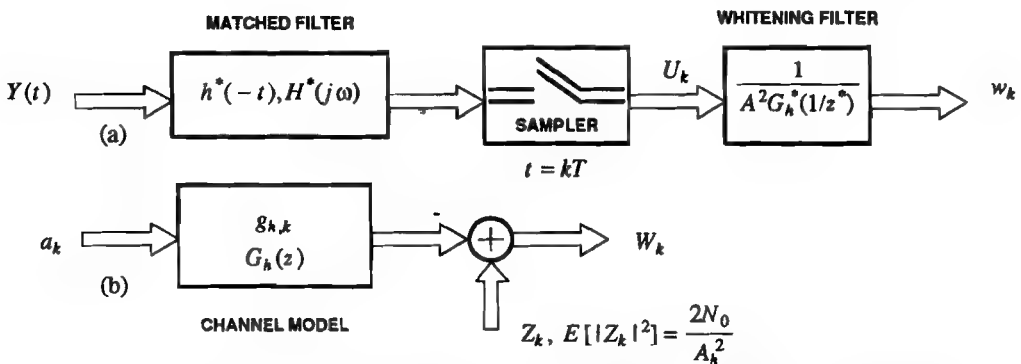
$$A_h^2 = \langle S_h \rangle_G. \quad (10.14)$$

$S_h(z)$  is the Z-transform of the pulse autocorrelation function  $\rho_h(k)$ , and is non-negative real-valued on the unit circle. It was shown in Section 9.3 that the WMF can be generalized to nonwhite channel noise, and that the equivalent channel model is essentially the same. In this chapter we will assume the channel noise is white.

The existence of the spectral factorization of (10.13) was established in Section 2.5 only for rational folded spectra. Fortunately, it holds more generally, in fact for any folded spectrum  $S_h(z)$  such that both  $S_h(e^{j\omega T})$  and  $\log S_h(e^{j\omega T})$  are integrable on the interval  $(-\pi/T, \pi/T)$ . This is known as the *Paley-Wiener condition*. From (10.14), the integrability of  $\log S_h(e^{j\omega T})$  is required for  $A_h^2$  to be well defined.

### Example 10-3.

If  $S_h(e^{j\omega T}) = 0$  over any interval of frequencies, then  $\log S_h(e^{j\omega T}) = -\infty$  over that same interval, and thus  $\log S_h(e^{j\omega T})$  is not integrable. Intuitively, this is obvious since for this condition the noise at the output of the sampled matched filter will have a vanishing power spectrum over an interval of frequencies, and clearly it cannot then be whitened.  $\square$



**Figure 10-1.** The whitened matched filter. (a) It consists of a matched filter, symbol-rate sampler, and maximum-phase whitening filter. (b) The equivalent channel model is a causal monic minimum-phase filter, with additive white and circularly symmetric Gaussian noise.

We cannot guarantee the existence of a spectral factorization of (10.13) unless the folded spectrum vanishes on the unit circle only at discrete points, and even then it must vanish in such a way that its logarithm is integrable. Fortunately, rational folded spectra can vanish at only a finite set of points on the unit circle (at most the number of zeros), and their logarithm is always integrable.

The existence of the WMF in Figure 10-1 depends on the spectral factorization of  $S_h(z)$ , which depends in turn on the integrability of  $S_h(e^{j\omega T})$ . The WMF exists for rational folded spectra without poles on the unit circle, but not for all non-rational folded spectra. The receiver design techniques described in this section do not apply to folded spectra for which the WMF does not exist.

Very useful for the sequel is the observation that the energy in the received pulse  $h(t)$  can be related to  $A_h$  and  $G_h(z)$  through

$$\rho_h(k) = A_h^2 \cdot g_{h,k} * g_{h,-k}^* \quad (10.15)$$

and hence

$$\sigma_h^2 = \rho_h(0) = A_h^2 \cdot \sum_{k=0}^{\infty} |g_{h,k}|^2. \quad (10.16)$$

In this section we will use the WMF of Figure 10-1a as the front end of the receiver. For purposes of designing the equalizer structures, it is appropriate to use the channel model of Figure 10-1b as the starting point.

### Probability of Error

It was shown in Section 8.2 that when a minimum-distance receiver design criterion is used for the equivalent channel model of Figure 10-1b, the error probability is approximately

$$P_e \approx K \cdot Q(1/2 \sqrt{\gamma}), \quad \gamma = d_{\min}^2 / \sigma^2, \quad (10.17)$$

where  $K \geq 1$  is the error coefficient, equal to the average number of signals at the minimum distance,  $d_{\min}$  is the minimum Euclidean distance between pairs of known signals, and  $\sigma^2 = N_0/A_h^2$  is the variance of the real or imaginary parts of the additive noise. The interpretation of the terminology *known signals* depends on the context. We will see two examples below: the isolated-pulse case, and the sequence case.

It was observed that the argument of  $Q(\cdot)$  has much greater impact on the error probability than the multiplicative constant  $K$ . Therefore, in this chapter we will compare modulation techniques primarily through their value of  $\gamma$ , rather than comparing the error probability directly. Since  $\gamma$  is the primary parameter of a particular receiver design impacting its probability of error, we will call it the *figure of merit* for the receiver equalization or detection technique. Generally, receivers with a higher figure of merit will have a lower error probability (although  $K$  must be taken in account to be definitive).

### Matched-Filter Bound

The WMF leads directly to two upper bounds on the figure of merit  $\gamma$  in the presence of ISI. The first bound is the matched filter bound, which presumes that a single data symbol is transmitted. That is, the input to the equivalent channel of Figure 10-1b is  $a_0 \cdot \delta_k$  for some isolated data symbol  $a_0$ . The output of the channel is then

$$W_k = a_0 g_{h,k} + Z_k. \quad (10.18)$$

This is the case of a received discrete-time signal in additive white Gaussian noise. The ML detector is shown in Section 9.3.1 to be a matched filter with transfer function  $G_h^*(1/z^*)$  followed by a sampler at  $t = 0$  and a slicer for the scaled data symbol  $\sigma_h \cdot a_0$ . Actually, this discrete-time matched filter is (within a constant) the inverse of the equalizer in the WMF, and thus the ML detector reverts to the output of the *continuous-time* matched filter  $H^*(j\omega)$  sampled at time  $t = 0$ . Since the continuous-time received signal in this case is  $a_0 \cdot h(t)$ , this is obvious in retrospect. The original matched filter was adequate without the need for an equalizer.

The set of known signals in (10.18) is  $a_0 \cdot g_{h,k}$  for all  $a_0$  in the symbol alphabet. The minimum distance for this set of known signals is

$$d_{\min}^2 = \min \sum_{k=0}^{\infty} |a_0^{(1)} g_{h,k} - a_0^{(2)} g_{h,k}|^2 = a_{\min}^2 \cdot \sum_{k=0}^{\infty} |g_{h,k}|^2, \quad (10.19)$$

where  $a_0^{(1)}$  and  $a_0^{(2)}$  are two different data symbols, the minimization is over all such pairs such that  $a_0^{(1)} \neq a_0^{(2)}$ , and  $a_{\min}$  is the minimum distance within the data-symbol alphabet. From (10.16), this can be written in the form

$$d_{\min}^2 = \frac{a_{\min}^2 \sigma_h^2}{A_h^2}. \quad (10.20)$$

Thus, from (10.17) we get a figure of merit of

$$\gamma_{MF} = \frac{d_{\min}^2}{\sigma^2} = \frac{a_{\min}^2 \sigma_h^2 / A_h^2}{N_0 / A_h^2} = a_{\min}^2 \cdot \frac{\sigma_h^2}{N_0}. \quad (10.21)$$

The constant  $\sigma_h^2 / N_0$  is a kind of signal-to-noise ratio, equal to the received pulse energy divided by the power spectrum of the white noise.

In the presence of ISI, the receiver will generally not achieve a figure of merit of  $\gamma_{MF}$  because the matched-filter receiver does not take ISI into account. However,  $\gamma_{MF}$  represents a very useful benchmark, since the difference between the actual receiver figure of merit (always smaller than  $\gamma_{MF}$ ) and  $\gamma_{MF}$  is a measure of the severity of the ISI, as well as the effectiveness of our methods for countering its effects.

### Figure of Merit for the MLSD

Another upper bound on the figure of merit is  $\gamma_{MLSD}$ , the figure of merit for the MLSD. No receiver based on equalization or other techniques for countering ISI can perform better than the MLSD. Thus,  $\gamma_{MLSD}$  represents an upper bound on the figure of merit. Moreover, any gap between  $\gamma_{MLSD}$  and  $\gamma_{MF}$  is due to ISI and not to

shortcomings in the receiver design. This difference is a fundamental measure of the severity of the ISI. Surprisingly, in many cases this difference is zero, meaning that within a multiplicative constant the error probability of the MLSD is essentially the same as the MF bound even in the presence of ISI.

The figure of merit for the MLSD is given by

$$\gamma_{\text{MLSD}} = \frac{d_{\min}^2}{N_0/A_h^2} = \frac{d_{\min}^2 A_h^2}{N_0}, \quad (10.22)$$

where

$$d_{\min}^2 = \min_{\substack{\{\epsilon_k, 1 \leq k \leq K\} \\ \epsilon_1 \neq 0}} \sum_{m=1}^{\infty} \left| \sum_{k=1}^K \epsilon_k g_{h,m-k} \right|^2. \quad (10.23)$$

The calculation of this minimum distance is described in Section 9.6.

A pair of very useful bounds on  $\gamma_{\text{MLSD}}$  can be developed simply. The first bound shows that the MLSD has a figure of merit less than or equal to the matched-filter bound. This follows from the simple observation that if we perform the minimization over a set of restricted error events, then this cannot match the minimum distance. In particular, in (10.23) constrain  $\epsilon_k = 0$  for  $2 \leq k \leq K$ , and then

$$\begin{aligned} d_{\min}^2 &\leq \min_{\epsilon_1 \neq 0} \sum_{m=1}^{\infty} |\epsilon_1 g_{h,m-1}|^2 \\ &= \min_{\epsilon_1 \neq 0} |\epsilon_1|^2 \sum_{k=0}^{\infty} |g_{h,k}|^2 = a_{\min}^2 \sum_{k=0}^{\infty} |g_{h,k}|^2 = a_{\min}^2 \sigma_h^2 / A_h^2, \end{aligned} \quad (10.24)$$

where we have used (10.16). Equation (10.24) leads to the desired upper bound on  $\gamma_{\text{MLSD}}$ ,

$$\gamma_{\text{MLSD}} \leq \frac{a_{\min}^2 \sigma_h^2 / A_h^2}{N_0 / A_h^2} = \frac{a_{\min}^2 \sigma_h^2}{N_0} = \gamma_{\text{MF}}. \quad (10.25)$$

It is possible for  $\gamma_{\text{MLSD}} = \gamma_{\text{MF}}$ , even in the presence of ISI. This will occur whenever one of the minimum-distance error events is the single error  $\epsilon_1$ , since in that case (10.24) becomes an equality.

#### Example 10-4.

In Example 8-11 the minimum-phase channel response is  $G_h(z) = 1 + \alpha z^{-1}$ . Assume the original data symbols are taken from the alphabet  $\{0,1\}$ , so that the error symbols have alphabet  $\{0,\pm 1\}$ . For the given  $G_h(z)$ , the error-event trellis has three states, as shown in Figure 10-2. Shown are the only two possible error events, and the shorter error event is *always* the minimum-distance event because its path metric is smaller by  $(1-\alpha)^2$ . The minimum distance error event has only a single non-zero error  $\epsilon_1$ . Hence,  $\gamma_{\text{MLSD}} = \gamma_{\text{MF}}$  for this channel, and the MLSD achieves the same figure of merit as the matched filter bound. We can verify this in another way, because  $d_{\min}^2 = 1 + \alpha^2$ , and hence

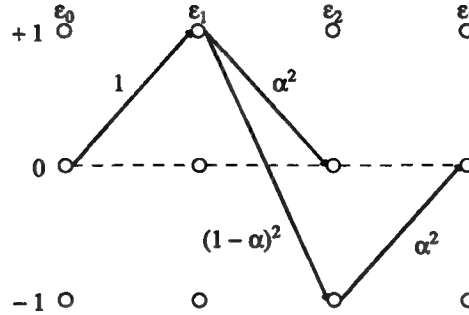


Figure 10-2. The trellis for determining the minimum distance in Example 10-4.

$$\gamma_{\text{MLSD}} = \frac{d_{\min}^2}{N_0/A_h^2} = \frac{1 + \alpha^2}{N_0} \cdot \frac{\sigma_h^2}{1 + \alpha^2} = \frac{\sigma_h^2}{N_0} = \gamma_{\text{MF}}. \quad (10.26)$$

In (10.26) we have used the fact that  $A_h^2 = \sigma_h^2/(1 + \alpha^2)$  and also that  $a_{\min} = 1$  for this data symbol alphabet.  $\square$

A lower bound on  $\gamma_{\text{MLSD}}$  follows from the inequality

$$\sum_{m=1}^{\infty} \left| \sum_{k=1}^K \epsilon_k g_{h,m-k} \right|^2 \geq \left| \sum_{k=1}^K \epsilon_k g_{h,1-k} \right|^2 = |\epsilon_1 g_{h,0}|^2 = |\epsilon_1|^2. \quad (10.27)$$

For any candidate error sequence, the left-hand side of (10.27) is greater than or equal to the right side. Thus, for the error sequence that minimizes the left side (and achieves  $d_{\min}^2$ ),  $|\epsilon_1|^2$  is not greater. If we minimize  $|\epsilon_1|^2$  over  $\epsilon_1$  (yielding  $a_{\min}^2$ ), we get an even smaller (or equal) quantity. Thus, it follows that,

$$d_{\min}^2 \geq a_{\min}^2, \quad \gamma_{\text{MLSD}} \geq \frac{a_{\min}^2 A_h^2}{N_0}. \quad (10.28)$$

This bound will prove useful in comparing the MLSD to other receivers later. (The lower bound is actually the figure of merit of the decision-feedback equalizer considered later.) If there is ISI ( $G_h(z) \neq 1$ ), and  $G_h(z)$  is an FIR filter (as it must be for application of the VA), then this inequality is strict (Problem 10-2).

### 10.1.2. Zero-Forcing Linear Equalizer

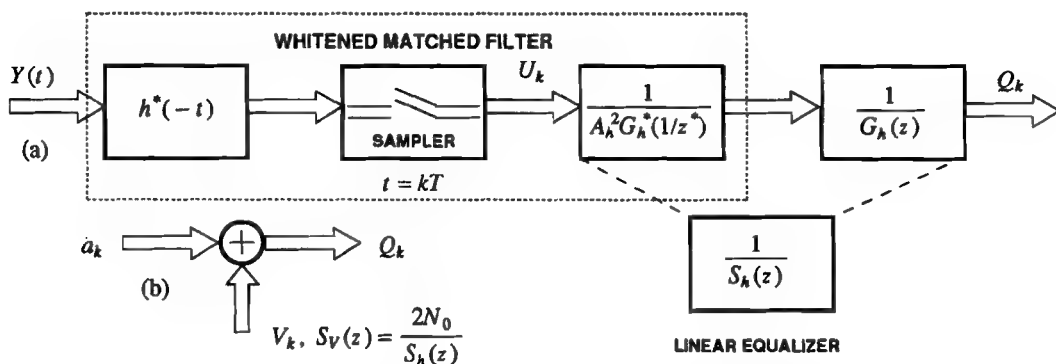
An obvious sub-optimal receiver eliminates ISI with an equalizer. This was done in Chapter 6, where the receive filter was chosen to satisfy the Nyquist criterion at the slicer input. In Section 8.3.1 the phenomenon of *noise enhancement* was observed, where in the process of amplifying to compensate for channel loss, the equalizer makes the noise variance larger at the slicer input.

The development in Chapter 6 left some open questions, however. We saw that the Nyquist criterion specifies neither the equalized pulse at the slicer input nor the receive filter. Even if the excess bandwidth of the equalized pulse is chosen, the pulse and the corresponding receive filter are not unique. At that time, we were unable to

specify the receive filter, from among those satisfying the Nyquist criterion, that minimizes the noise at the slicer or equivalently minimizes the probability of symbol error. We are in a position to do that now.

The optimal receive filter under the constraint that the Nyquist criterion must be satisfied at the slicer input can now be determined easily because the WMF can be used as a front end for *any* criterion of optimality, including this one. Placing a discrete-time symbol-rate filter at the WMF output, the filter that will satisfy the Nyquist criterion is unique because of the symbol-rate sampling. That filter, called the *zero-forcing linear equalizer (LE-ZF)*, is the inverse filter  $G_h^{-1}(z)$ , assuming the discrete-time equivalent channel model for the WMF of Figure 10-1b, since the Nyquist criterion is equivalent to an overall unity transfer function for the WMF plus equalizer. The term "zero-forcing" refers to the fact that we are constraining the ISI to be entirely absent, a constraint that will be relaxed in Section 10.2. Since  $G_h(z)$  is a monic causal minimum-phase filter, the equalizer must have all its poles inside or on the unit circle; hence it is a causal filter, although it is not necessarily minimum phase or stable, because of the possibility of poles on the unit circle. We will see that when  $G_h(z)$  has zeros on the unit circle, the LE-ZF is not useful since the noise at the slicer input would have to have infinite variance (the figure of merit would be zero).

Combining the equalizer  $G_h^{-1}(z)$  with the WMF discrete-time equalizer in Figure 10-1a, we show the resulting LE receiver structure in Figure 10-3. Not shown is the demodulator that may precede the matched filter. Among all receiver structures consisting of a demodulator, receiver filter, and equalizer constrained to eliminate ISI at the slicer input, this one is optimal (minimizes error probability). The combined discrete-time equalizer has transfer function  $S_h^{-1}(z)$ , the inverse of the folded spectrum.



**Figure 10-3.** The zero-forcing linear equalizer. (a) It consists of a matched filter, symbol-rate sampler, and a discrete-time equalizer with transfer function that is the inverse of the folded spectrum. (b) The equivalent channel model consists of an ideal channel plus nonwhite Gaussian noise.

### Figure of Merit for the LE-ZF

The figure of merit for the LE is easily determined from the fact that the noise at the WMF output is white and circularly symmetric with variance  $2N_0/A_h^2$ . The power spectrum of the noise at the output of the LE is therefore (suppressing the  $z$  and  $1/z^*$  variables)

$$S_V = \frac{2N_0}{A_h^2} \cdot \frac{1}{G_h} \cdot \frac{1}{G_h^*} = \frac{2N_0}{S_h}. \quad (10.29)$$

The input to the slicer is the current data symbol plus an additive Gaussian noise sample with variance

$$\sigma_V^2 = 2\sigma^2 = 2N_0 \cdot \langle S_h^{-1} \rangle_A. \quad (10.30)$$

The minimum distance is that of the data symbol,  $a_{\min}$ , and thus the figure of merit is

$$\gamma_{\text{LE-ZF}} = \frac{a_{\min}^2}{N_0} \cdot \langle S_h^{-1} \rangle_A^{-1}. \quad (10.31)$$

For rational spectra, it is easier to calculate  $\gamma_{\text{LE-ZF}}$  by finding the coefficient of  $z^0$  in  $S_h^{-1}(z)$  than it is to evaluate this integral.

#### Example 10-5.

For the first-order all-pole received pulse  $h(t) = \sigma_h \sqrt{2a} e^{-aT} u(t)$  of Example 7-10,

$$S_h^{-1}(z) = \frac{(1 - \alpha z^{-1})(1 - \alpha z)}{(1 - \alpha^2)\sigma_h^2} \quad (10.32)$$

so that the coefficient of  $z^0$  is  $(1 + \alpha^2)(1 - \alpha^2)^{-1}\sigma_h^{-2}$ . Thus, the figure of merit is

$$\gamma_{\text{LE-ZF}} = a_{\min}^2 \cdot \frac{\sigma_h^2}{N_0} \cdot \frac{1 - \alpha^2}{1 + \alpha^2} = \gamma_{\text{MF}} \cdot \frac{1 - \alpha^2}{1 + \alpha^2}. \quad (10.33)$$

Note that  $\gamma_{\text{LE-ZF}} \rightarrow 0$  as  $|\alpha| \rightarrow 1$ , which is the case where the channel pole approaches the unit circle.  $\square$

#### Example 10-6.

For the first-order all-zero received pulse  $h(t) = h_0(t) + \alpha h_0(t - T)$  of Example 7-11,

$$S_h^{-1}(z) = \frac{1 + \alpha^2}{\sigma_h^2} \cdot \frac{1}{(1 - \alpha z^{-1})(1 + \alpha z)} \quad (10.34)$$

and when expanded in  $z^{-k}$  has a coefficient of  $z^0$  of  $(1 + \alpha^2)(1 - \alpha^2)^{-1}\sigma_h^{-2}$ . As a result, the figure of merit is

$$\gamma_{\text{LE-ZF}} = a_{\min}^2 \cdot \frac{\sigma_h^2}{N_0} \cdot \frac{1 - \alpha^2}{1 + \alpha^2}. \quad (10.35)$$

Note that  $\gamma_{\text{LE-ZF}} \rightarrow 0$  as  $|\alpha| \rightarrow 1$ , because the channel zero approaches the unit circle and the LE-ZF cannot equalize it.  $\square$

### Bound on Figure of Merit

An upper bound on  $\gamma_{\text{LE-ZF}}$  that is useful for later comparisons can be developed. Let the equalizer  $G_h^{-1}(z)$  have impulse response  $f_{h,k}$ . This response is causal, and since  $G_h^{-1}(\infty) = 1$ , it is also monic ( $f_0 = 1$ ). Since the input noise to this LE-ZF is white, we can express the output variance in terms of this impulse response as

$$\sigma_V^2 = \frac{2N_0}{A_h^2} \cdot \sum_{k=0}^{\infty} |f_{h,k}|^2 \geq \frac{2N_0}{A_h^2}, \quad (10.36)$$

with equality if and only if  $G_h(z) = 1$  (there is no ISI). From this it follows that

$$\gamma_{\text{LE-ZF}} = \frac{a_{\min}^2}{\frac{1}{2} \cdot \sigma_V^2} \leq \frac{a_{\min}^2 A_h^2}{N_0}. \quad (10.37)$$

The right side of (10.37) is the figure of merit for the zero-forcing decision-feedback equalizer, considered later in Section 10.1.3.

### Existence of the LE-ZF

The existence of the LE-ZF is not guaranteed, but depends on the folded spectrum  $S_h(z)$ . The first requirement is the existence of the WMF, or in other words that  $\log S_h(e^{j\omega T})$  be integrable. As we saw earlier, the WMF is precluded if, for example,  $S_h(e^{j\omega T})$  vanishes on an interval. The LE-ZF has the more stringent requirement that the filter  $G_h^{-1}(z)$  be stable, and that the noise variance at its output be finite. The problematic case is where  $G_h(z)$  has one or more zeros on the unit circle, in which case  $G_h^{-1}(z)$  is a well-defined filter but is not stable and the variance of the noise at the slicer is infinite. This can be seen from (10.36), because when  $G_h^{-1}(z)$  has a pole on the unit circle,  $f_{h,k}$  does not decay to zero as  $k \rightarrow \infty$ , and thus  $\sum_{k=0}^{\infty} |f_{h,k}|^2 = \infty$ .

In summary, when  $S_h(z)$  is rational, the LE-ZF will not be useful (in the sense that the slicer noise variance would be infinite) whenever  $S_h(z)$  has zeros on the unit circle. Intuitively, this is because the LE-ZF, in the process of adding gain to compensate for channel attenuation, cannot compensate for even an algebraic zero in the frequency response of the channel. Note that under these conditions the WMF *does* exist, so that for example the MLSD can be implemented.

When  $S_h(z)$  is not rational, the precise condition for the LE-ZF to be useful is that  $S_h^{-1}(e^{j\omega T})$  be integrable, which guarantees a finite noise variance at the slicer input, from (10.30).

### 10.1.3. Zero-Forcing Decision-Feedback Equalizer

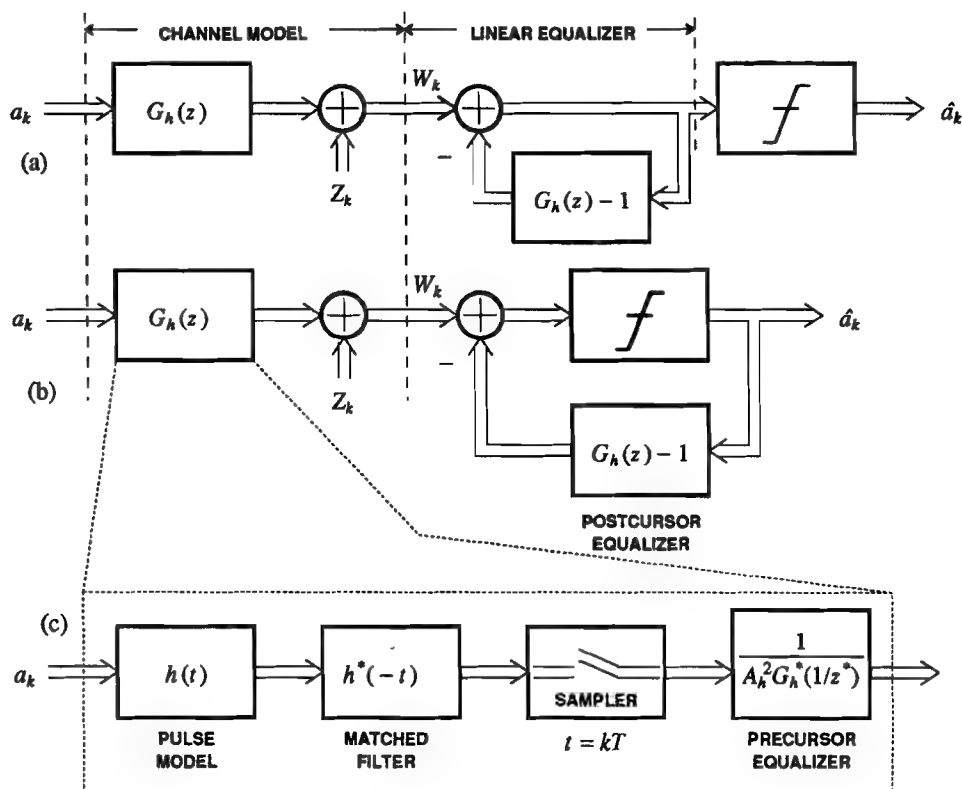
The *zero-forcing decision-feedback equalizer (DFE-ZF)* is a nonlinear receiver structure that offers a performance intermediate between the LE and the MLSD. In the absence of channel coding (Chapter 14), the DFE is frequently used since it offers a good compromise between performance and implementation complexity.

The DFE-ZF follows from the observation that the WMF equivalent channel model of Figure 10-1b is monic and causal. Thus, the residual ISI at this point is



called *postcursor ISI*, meaning that the interference at the WMF output is only from *past* data symbols. This is distinguished from *precursor ISI*, where the interference is from future symbols. The distinguishing feature of postcursor ISI is that if we know the past data symbols, we can *cancel* the ISI by subtracting a replica of the ISI from the WMF output. If we are making symbol-by-symbol decisions using a slicer, then in fact we do have *estimates* of the past data symbols, namely the slicer output. In the DFE, these estimates are used to cancel the postcursor ISI at the slicer input.

The DFE is related to the LE in Figure 10-4. In Figure 10-4a, the LE is shown in a different form, where the equalizer filter  $G_h^{-1}(z)$  is realized as a filter  $G_h(z) - 1$  placed in a feedback loop. It is easily shown that the transfer function of this feedback system is  $G_h^{-1}(z)$ . The feedback loop is also realizable; because  $G_h(z)$  is causal and

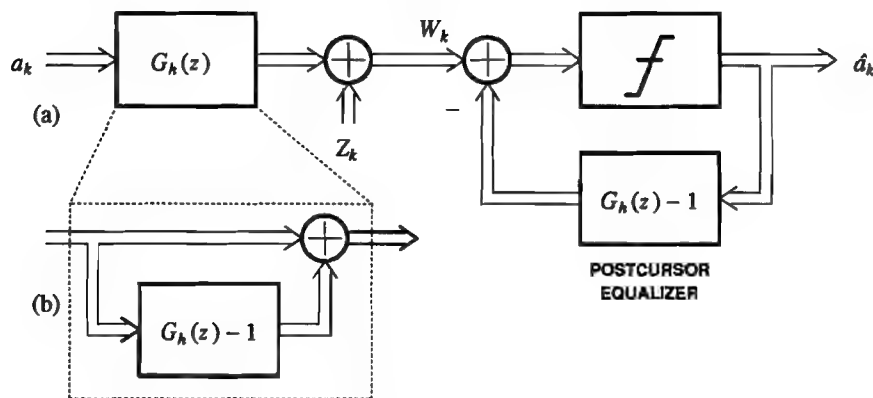


**Figure 10-4.** The zero-forcing decision-feedback equalizer (DFE-ZF). (a) A different realization of the LE-ZF, in which the filter  $G_h^{-1}(z)$  is implemented with a feedback filter. (b) The DFE-ZF, in which the slicer is moved inside the feedback loop, reducing the noise enhancement. (c) Expansion of the channel model. The front end of the DFE-ZF, identical to the WMF, consists of a matched filter and precursor equalizer, where the latter also doubles as a whitening filter.

monic,  $G_h(z) - 1$  is a strictly causal response with no zero-delay tap (such a tap could not be realized in the feedback loop!). This filter is stable as long as  $G_h(z)$  has no zeros on the unit circle (it is strictly minimum-phase).

The DFE, first suggested by Austin [1,2], is shown in Figure 10-4b. There are two equivalent ways of looking at this receiver structure. One follows from the observation that, in the absence of noise, the output of the LE (input to the slicer) in Figure 10-4a is precisely the data symbol  $a_k$ . This property is not affected by moving the slicer *inside* the feedback loop, since in the absence of noise the slicer has no effect. The effect on the *noise* of moving the slicer inside the feedback loop is beneficial, however, since the effect of the slicer is to remove any noise that would otherwise recirculate back through the feedback loop to the input to the slicer (this will be verified mathematically momentarily). Moving the slicer inside the feedback loop also has a beneficial stabilizing effect; even if  $G(z)$  has zeros on the unit circle and the LE-ZF filter is not stable, the nonlinear filter of Figure 10-4b is stable, since the output of the slicer is always bounded! Thus, the DFE-ZF exists and is a stable system whenever the WMF exists.

The second viewpoint, ISI cancellation, is illustrated in Figure 10-5. The channel model of Figure 10-4c is shown in a different way in Figure 10-5b. As illustrated in Figure 10-4c, the front end of the DFE-ZF is identical to the WMF, including a matched filter, symbol-rate sampler, and maximum-phase whitening filter (labeled *precursor equalizer*). In the context of the DFE-ZF, that whitening filter plays another, more important role, that of equalizing the response to be causal. This is why it is also called a precursor equalizer. If there is any ISI at all, then there is precursor ISI because of the symmetry about  $t = 0$  of the matched filter response. The precursor ISI is eliminated by the precursor equalizer. The resulting model  $G_h(z)$  is monic and causal, and hence can be viewed as shown in Figure 10-5b. Since the output of the



**Figure 10-5.** (a) Another view of the zero-forcing decision-feedback equalizer (DFE-ZF). (b) An expansion of the channel model that shows clearly that this model introduces postcursor ISI that is cancelled by the DFE.  $G_h(z)$  is causal and monic, so  $G_h(z) - 1$  is strictly causal.

slicer is an estimate  $\hat{a}_k$  of the current data symbol  $a_k$ , if  $\hat{a}_k = a_k$ , then the postcursor equalizer exactly cancels the ISI introduced by the model in Figure 10-5b.

An example is shown in Figure 10-6. The output of the sampled matched filter has both precursor and postcursor ISI, but the precursor equalizer eliminates the precursor ISI. The postcursor ISI is cancelled by the *postcursor equalizer*  $G_h(z) - 1$  using the symbol estimates generated by the slicer. Since the ISI at the output of the precursor equalizer has transfer function  $G_h(z) - 1$ , a replica of this ISI can be generated using the feedback filter. This argument depends on the assumption that all decisions are correct. In fact, when the slicer makes incorrect decisions, the ISI correction becomes flawed for future decisions. This phenomenon is known as *error propagation*, and is discussed later.

### Figure of Merit of the DFE-ZF

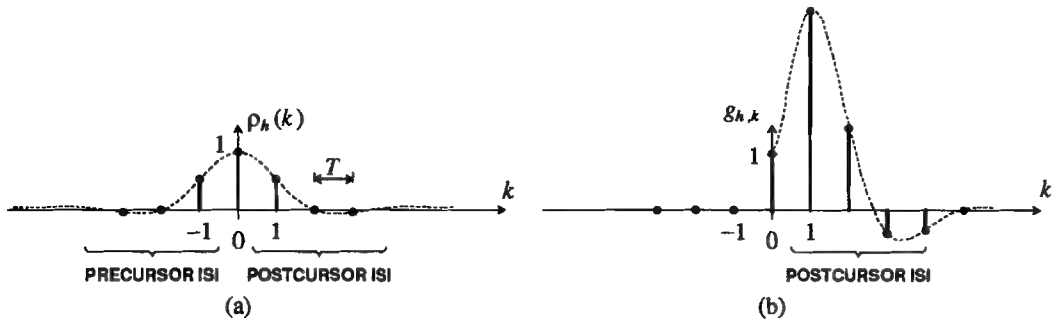
The DFE performance is easily calculated assuming that all past decisions are correct. In that case, the signal component at the slicer input is the current data symbol  $a_k$ , free of ISI, and hence the minimum distance is the data-symbol alphabet minimum distance  $a_{\min}$ . The noise at the slicer input is precisely the noise at the WMF output, which has variance  $2N_0/A_h^2$ . The figure of merit is therefore

$$\gamma_{\text{DFE-ZF}} = \frac{a_{\min}^2}{N_0/A_h^2} = \frac{a_{\min}^2 A_h^2}{N_0}. \quad (10.38)$$

Using (10.14) this can be expressed directly in terms of the folded spectrum as

$$\gamma_{\text{DFE-ZF}} = \frac{a_{\min}^2}{N_0} \cdot \langle S_h \rangle_G = \frac{a_{\min}^2}{N_0} \cdot \langle S_h^{-1} \rangle_G^{-1}. \quad (10.39)$$

The second form of (10.39) is written in a form to emphasize the comparison to (10.31); the two equations are similar except that the arithmetic mean in (10.31) is



**Figure 10-6.** Example pulse shapes in a DFE. (a) The pulse shape at the output of the sampled matched filter is always symmetric about  $k = 0$ . (b) After the precursor equalizer (whitening filter) the pulse is causal and minimum phase; that is, it has postcursor ISI only. The postcursor equalizer cancels the tail.

replaced by the geometric mean in (10.39). For rational spectra,  $\gamma_{\text{DFE-ZF}}$  is easily calculated directly from the spectrum without using (10.39). However, (10.39) is very useful for non-rational spectra, or to avoid performing a spectral factorization.

#### Example 10-7.

Repeating the first-order all-pole pulse in Example 10-5 for the DFE-ZF,  $A_h^2 = \sigma_h^2(1 - \alpha^2)$ . This illustrates the ease with which  $A_h^2$  can be calculated without evaluating a geometric mean integral. Then the figure of merit is

$$\gamma_{\text{DFE-ZF}} = a_{\min}^2 \cdot \frac{\sigma_h^2}{N_0} (1 - \alpha^2). \quad (10.40)$$

As in the LE-ZF,  $\gamma_{\text{DFE-ZF}} \rightarrow 0$  as  $|\alpha| \rightarrow 1$ , because the channel pole is approaching the unit circle. Also, from Example 10-5,  $\gamma_{\text{DFE-ZF}}/\gamma_{\text{LE-ZF}} = 1 + \alpha^2$ , so that as expected the DFE-ZF always has a larger figure of merit than the LE-ZF, by as much as 3 dB.  $\square$

#### Example 10-8.

Repeating the first-order all-zero pulse of Example 10-6 for the DFE-ZF, in this case by inspection  $A_h^2 = \sigma_h^2/(1 + \alpha^2)$ , and hence the figure of merit is

$$\gamma_{\text{DFE-ZF}} = a_{\min}^2 \cdot \frac{\sigma_h^2}{N_0} \cdot \frac{1}{1 + \alpha^2}. \quad (10.41)$$

In contrast to the LE-ZF, the DFE-ZF is well behaved as  $|\alpha| \rightarrow 1$ . The DFE-ZF suffers at most a 3 dB penalty relative to the MF bound, whereas the LE-ZF may suffer an arbitrarily large penalty.  $\square$

Two bounds, (10.28) and (10.37), establish that the figure of merit of the DFE-ZF falls between the LE-ZF and the MLSD,

$$\gamma_{\text{LE-ZF}} \leq \gamma_{\text{DFE-ZF}} \leq \gamma_{\text{MLSD}}. \quad (10.42)$$

These are strict inequalities, unless there is no ISI, in which case they become equalities. (An additional inequality is that  $\gamma_{\text{MLSD}} \leq \gamma_{\text{MF}}$ , which can be an equality even in the presence of ISI.) The intuitive reason the DFE-ZF performs better than the LE-ZF is that postcursor ISI is cancelled without noise enhancement, since the slicer removes noise fed back through the postcursor equalizer filter.

### Existence of the DFE-ZF

Since the DFE-ZF uses the WMF as a front end, it exists whenever the WMF exists; that is, both  $S_h(e^{j\omega T})$  and  $\log S_h(e^{j\omega T})$  must be integrable. Roughly speaking, the folded spectrum  $S_h(e^{j\omega T})$  cannot vanish on an interval, although algebraic zeros (zeros characteristic of rational spectra) are permissible. Thus, another advantage of the DFE-ZF over the LE-ZF is that the DFE-ZF has a finite noise variance at the slicer input even when the folded spectrum has zeros on the unit circle, whereas the slicer input noise will have infinite variance for the LE-ZF under the same conditions. This is a desirable side effect of performing precursor equalization, but not postcursor equalization, using a linear filter.

### Optimality of the WMF as the DFE-ZF Front End

We based the DFE on the WMF, without considering alternative filters. The basic concept of canceling postcursor ISI with a postcursor equalizer does not depend on the details of the impulse response of the precursor equalizer, except that the equivalent discrete-time isolated-pulse response up to and including the precursor equalizer must be monic and causal, so that the ISI is in fact postcursor.

As shown in Figure 10-7, any filter  $E(z)$  can be placed at the output of the WMF, changing the equivalent transfer function from  $G_h(z)$  to  $G_h(z)E(z)$ , as long as the feedback filter is changed from  $G_h(z) - 1$  to  $G_h(z)E(z) - 1$ .  $E(z)$  must be chosen to meet the two constraints above:

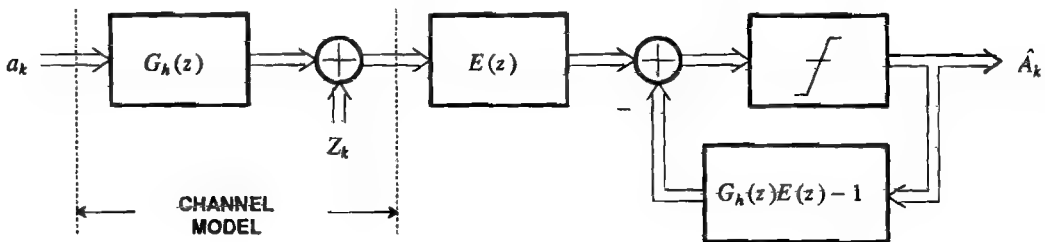
- Since  $G_h(z)$  is causal,  $G_h(z)E(z)$  will be causal if and only if  $E(z)$  is causal.
- Since  $G_h(z)$  is monic ( $G_h(\infty) = 1$ ),  $G_h(z)E(z)$  will be monic ( $G_h(\infty)E(\infty) = 1$ ) if and only if  $E(z)$  is monic ( $E(\infty) = 1$ ).

For a filter  $E(z)$  with impulse response  $e_k$  meeting these constraints, since the noise at the input to  $E(z)$  is white, the variance of the output (input noise to the slicer) will have variance

$$\frac{2N_0}{A_h^2} \cdot \sum_{k=0}^{\infty} |e_k|^2 \geq \frac{2N_0}{A_h^2} \quad (10.43)$$

with equality if and only if  $E(z) = 1$ . This establishes that the original configuration of Figure 10-4 is optimal in the sense of minimizing the noise variance at the slicer input.

In Section 7.3.2, it was pointed out that the minimum-distance receiver design (which is equivalent to the MLSD) does not require a minimum-phase spectral factorization. Equivalently, adding an allpass filter to the WMF and to the equivalent channel model will not change the minimum-distance criterion. This would be equivalent to making  $E(z)$  a causal monic filter with a constant magnitude response in Figure 10-7. We have just shown, however, that the minimum-phase spectral factorization is critical to the DFE, because it minimizes the noise at the slicer input. Adding a monic



**Figure 10-7.** An arbitrary DFE-ZF precursor equalizer can be realized by concatenating a monic and causal filter  $E(z)$  with the WMF front end.

and causal filter  $E(z)$  with constant magnitude  $|E(e^{j\omega T})|$  to the WMF will increase the noise variance without affecting the signal level at the slicer.

An intuitive explanation for this difference between the MLSD and the DFE is as follows: the DFE relies on the first sample of the equivalent impulse response to make the decision on the data symbol, and throws away the remainder of the received signal energy in the postcursor ISI. The minimum-phase channel model maximizes the energy in the first sample (see Problem 2-23), and is thus important to the performance of the DFE. The MLSD, on the other hand, uses *all* the energy in the equivalent channel impulse response, and hence will not be affected by the relative distribution of that energy between the first sample and the postcursor ISI.

In actual practice, the minimum-phase spectral factorization is helpful in the implementation of the MLSD as well. As shown in Section 9.5, the use of the Viterbi algorithm to realize the DFE requires that the channel response be FIR, and the computational complexity of the algorithm increases exponentially with the number of taps in the FIR channel model. It is often necessary to approximate an IIR response  $G_h(z)$  with an FIR filter, and the number of taps required for a good approximation will generally be minimized when the spectral factorization is minimum-phase, since again that concentrates the energy in the low-delay coefficients.

### Linear Predictor Interpretation of DFE-ZF

The optimal DFE-ZF structure can be derived in a slightly different way, one which lends additional insight. This alternative derivation illustrates a connection between optimal linear prediction (Section 3.2.3) and the WMF and DFE, and also explains in another way why the LE results in more noise enhancement than the DFE. The connection is illustrated in Figure 10-8, where a DFE is placed at the output of a LE-ZF rather than WMF, and consists of a linear predictor  $E(z)$ , which introduces postcursor ISI, and a DFE postcursor equalizer. The key observation is that the noise at the output of the LE-ZF is not white, as demonstrated by (10.29), unless of course there is no ISI ( $G_h(z) = 1$ ). Since the noise samples are correlated, we can take advantage of this correlation, and apply an optimal linear prediction error filter  $E(z)$  as shown in Figure 10-8. Like all linear prediction error filters,  $E(z)$  is monic and causal. Thus, while  $E(z)$  introduces ISI (which was absent at the output of the LE-ZF), this is postcursor ISI and can be canceled by a DFE-ZF feedback filter without enhancing the noise.

As shown in Section 3.2.3, the optimal  $E(z)$  is the monic minimum-phase whitening filter. The noise power spectrum at the input to the predictor is proportional to  $1/G_h(z)G_h^*(1/z^*)$ , and the monic minimum-phase portion is thus  $G_h^{-1}(z)$ . The optimal whitening filter is the inverse of this,  $G_h(z)$ , which also happens to be the inverse of the LE-ZF filter  $G_h^{-1}(z)$ . Thus, the optimal predictor and the LE-ZF filter cancel one another, the optimal front end for the DFE is the WMF, and the optimal postcursor equalizer is  $E(z) - 1 = G_h(z) - 1$ .

This establishes in a different way that the WMF is a front end that generates white noise at its output, and among all such front ends is the one that minimizes the noise variance. It also explains in a compelling way the enhanced performance of the DFE-ZF relative to the LE-ZF. The DFE-ZF takes advantage of the correlation of

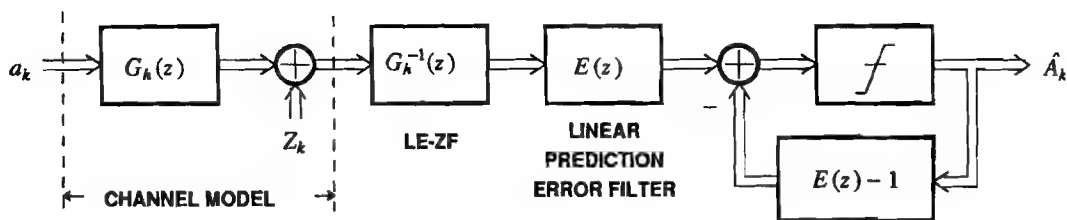


Figure 10-8. Showing the connection between the optimal DFE precursor equalizer and optimal linear prediction. The DFE precursor equalizer can be obtained by applying a linear predictor to the nonwhite noise at the LE-ZF output.

noise samples at the output of the LE-ZF to reduce the noise variance at the slicer input, and cancels the resulting postcursor ISI using a DFE postcursor equalizer.

### DFE Error Propagation

One obvious potential problem with the DFE is that any decision errors at the output of the slicer will cause a corrupted estimate of the postcursor ISI to be generated by the postcursor equalizer. The result is that a single error causes a reduction in the margin against noise for a number of future decisions. This phenomenon is called *error propagation*, and results in an error rate greater than would be predicted on the basis of SNR calculations alone. Error propagation is explored further in Appendix 10-A, where it is shown that the benefit of reduced noise enhancement usually far outweighs the effect of error propagation.

#### 10.1.4. Transmitter Precoding

DFE error propagation can be avoided by using *transmitter precoding*. Transmitter precoding is sometimes called *Tomlinson-Harashima coding*, in honor of its co-inventors [3,4,5]. This technique has also been called *generalized partial response* [6], since the ordinary partial response invented earlier [7] is a special case (Chapter 12).

The idea of precoding is to move the cancellation of the postcursor ISI to the transmitter, where the past transmitted symbols are known without the possibility of errors. However, this means that the postcursor ISI impulse response  $G_h(z)$  must be known precisely at the transmitter. In practice, in most situations this impulse response must be estimated in the receiver using adaptive filter techniques (Chapter 11), and passed back to the transmitter in order to use transmitter precoding. This is feasible on channels that are time-invariant or slowly time-varying, but is not feasible on channels (such as mobile radio) that are rapidly time-varying.

The basis of transmitter precoding is the observation that the channel model through the WMF,  $G_h(z)$ , and LE equalizer  $G_h^{-1}(z)$ , both linear and time-invariant, can be reversed without compromising the requirement that the Nyquist criterion be satisfied at the slicer input, as shown in Figure 10-9a. The LE equalizer  $G_h(z)$  can be put in the transmitter. There are two benefits to this:

- Since the receiver is now the WMF followed directly by a slicer, the noise at the slicer input is that of the WMF; that is, it is the same as for the DFE. Thus, even though the receiver uses linear equalization, it does not suffer the noise enhancement of linear equalization because the equalization is done *prior* to the channel, where the noise is introduced.
- The error probability may actually be slightly lower than with the DFE, because the postcursor ISI cancellation is done in the transmitter, and there is no possibility of error propagation as there is in the DFE.

In the transmitter linear equalization of Figure 10-9a, the transmitted data symbols  $x_k$  are the original data symbols  $a_k$  filtered by  $G_h^{-1}(z)$ . This filter is simply placed between the original data symbols and the pulse-amplitude modulator, so that the transmitted signal becomes, for transmit pulse shape  $g(t)$ ,  $\sum_{k=-\infty}^{\infty} x_k g(t - kT)$ .

When the transmit filter, channel, and WMF front end in the receiver are taken into account, the channel model of Figure 10-9a results.

Simply doing the equalization in the transmitter as shown in Figure 10-9a, is not practical, however, because it increases the average and peak power in the transmitted signal. If the impulse response of  $G_h^{-1}(z)$  is  $f_{h,k}$ , which is causal and monic, then the peak transmitted sample is increased by a factor of  $\sum_{k=0}^{\infty} |f_{h,k}| > 1$ . Likewise, if the data symbols are independent and identically distributed, then the average power of the transmitted symbols is multiplied by a factor  $\sum_{k=0}^{\infty} |f_{h,k}|^2 > 1$ . If we penalize the system for these increases in transmitted peak and average power, a penalty which increases with the severity of the ISI, then the DFE will usually turn out to be superior.

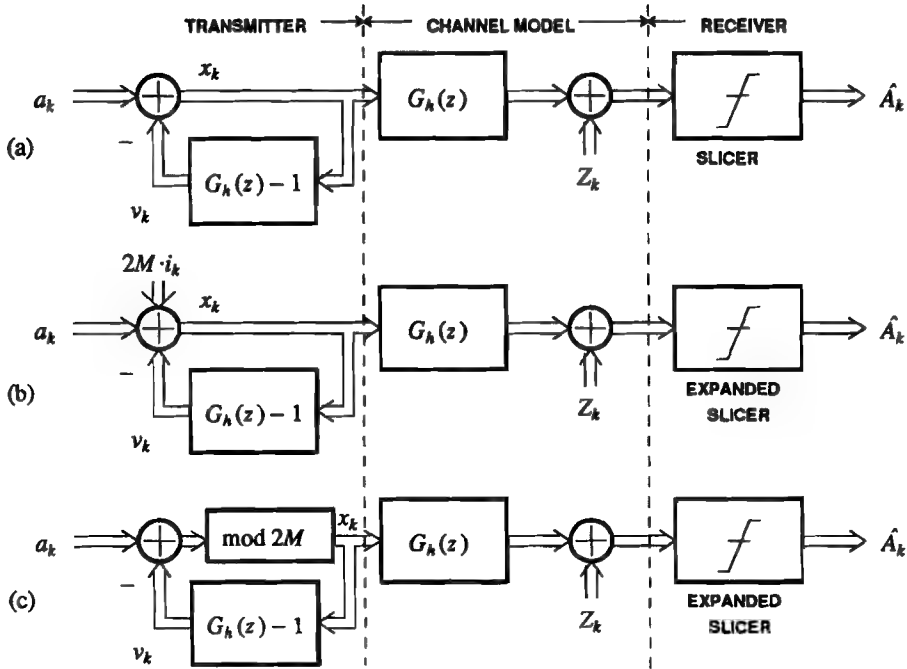
### Modification to Reduce the Transmitted Power

Fortunately, there is a simple solution that substantially reduces these peak and average power penalties, and in fact makes them go away entirely in the limit of large data-symbol constellations. This approach is easiest to understand in the one-dimensional case, so assume the data symbols  $a_k$  are drawn from the  $M$ -ary alphabet (where  $M$  is even)  $\{-(M-1), -(M-3), \dots, -3, -1, 1, 3, \dots, (M-3), (M-1)\}$ . That is, the data symbols are chosen among all odd integers in the range  $(-M, M)$ . Consider the modification of the transmitter-equalizer shown in Figure 10-9b, in which an additional term  $2M \cdot i_k$  is added to the feedback, where the sequence of integers  $\{i_k, -\infty < k < \infty\}$  is yet to be determined. Define an *expanded symbol*  $c_k$ ,

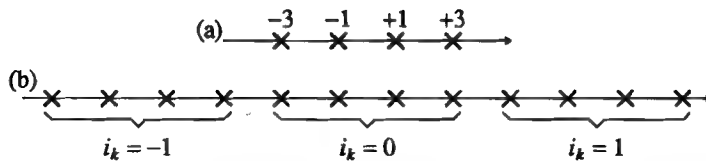
$$c_k = a_k + 2M \cdot i_k. \quad (10.44)$$

Since  $a_k$  is an odd integer and  $2M \cdot i_k$  is an even integer, their sum is odd, and the alphabet of the expanded symbol  $c_k$  is the set of *all* odd integers, not those limited to the range  $(-M, M)$ . Both the original data symbol alphabet and the expanded data symbol alphabet are illustrated in Figure 10-10. The original data symbol  $a_k$  can be recovered from the expanded symbol  $c_k$  by reducing it modulo  $2M$ ,





**Figure 10-9.** Derivation of the transmitter precoder. (a) The LE-ZF, realized as a feedback filter as in Figure 10-4a, can be placed in the transmitter rather than the receiver. (b) For the data-symbol alphabet specified in the text, an arbitrary sequence of samples  $2M \cdot i_k$  can be added to the data symbols, as long as the receiver slicer is appropriately expanded. (c) The transmitted power can be minimized by choosing  $2M \cdot i_k$  to yield an equivalent modulo  $2M$  operation.



**Figure 10-10.** Data symbol alphabets for transmitter precoding: (a) The original data symbols  $a_k$  for  $M = 4$ , and (b) the expanded data symbols  $c_k$ , shown for three values of  $i_k$ . (In general,  $i_k$  can assume larger values as well.)

$$a_k = c_k \text{ modulo } 2M . \quad (10.45)$$

By  $(x \text{ modulo } 2M)$ , we mean specifically the unique value of  $x + 2M \cdot m$  when integer  $m$  is chosen such that  $-M < x + 2M \cdot m \leq M$ . The *precoded symbol*  $x_k$  is transmitted, where

$$x_k = a_k + 2M \cdot i_k - v_k . \quad (10.46)$$

Figure 10-9b is equivalent to Figure 10-9a with the data symbol  $a_k$  replaced by the expanded symbol  $c_k$  of (10.44). Thus, in the receiver, the data symbol  $a_k$  can be recovered by first applying an *expanded slicer*, which detects the expanded symbol  $c_k$ , and then reducing the result modulo  $2M$ . The expanded slicer simply applies thresholds to detect a data symbol expected to be any odd integer, as opposed to the original slicer that expects an odd integer in the range  $(-M, M)$ . With the expanded slicer, the system of Figure 10-9b works just like Figure 10-9a for any sequence of integers  $i_k$ . Furthermore, the error probability is essentially the same, since the minimum distance is unchanged (the distance between odd integers is  $d_{\min} = 2$  in either case). There will be a slight increase in error probability due to end effects, as the original constellation is bounded, and the two outer symbols are therefore detected with a slightly lower error probability, a property lost with the expanded constellation.

The final step is to choose the sequence of integers  $i_k$ . For this purpose, observe from Figure 10-9b and (10.46) that a term  $2M \cdot i_k$  appears directly in the precoded symbols  $x_k$ . Since our original goal was to reduce the peak or average power penalty,  $i_k$  should be chosen to minimize the magnitude of each  $x_k$ . In fact  $i_k$  can always be chosen to limit  $x_k$  to the range  $(-M, M]$ , which is equivalent to reducing it modulo  $2M$ ; choosing any other  $i_k$  results in a precoded symbol with a larger magnitude. This transmitter precoding approach, shown in Figure 10-9c, results in a small increase in the range of the precoded symbol alphabet, as the original data symbols are limited in magnitude to  $M-1$ , and the precoded symbols are limited to  $M$ . For large  $M$ , this is only a slight increase in transmitted power.

The operation of the precoder can be better understood with the following simple (and accurate, for large  $M$ ) model that predicts the statistics of the precoded symbols for an idealized statistical model of the original data symbols. Assume that the data symbols  $A_k$  are independent uniformly distributed random variables on  $(-M, M]$ . Of course, they actually have a discrete distribution, but this approximation becomes more accurate as  $M$  increases. This approximation is an important analytical tool in coding theory, and has been called the *continuous approximation* [8]. We will now show that under these conditions, the precoded symbols  $X_k$  are also independent and uniform-distributed on  $(-M, M]$ . First, in Figure 10-9c,

$$X_k = (A_k - V_k) \text{ modulo } 2M. \quad (10.47)$$

The first observation is that  $X_k$  is uniformly distributed and independent of  $V_k$ , regardless of the channel response  $G_h(z)$ .

### Exercise 10-1.

Show that  $P_{X_k|V_k}(x_k|v_k)$  is a uniform distribution on  $(-M, M]$ . Since  $P_{X_k|V_k}(x_k|v_k)$  is not a function only of  $v_k$ ,  $X_k$  is statistically independent of  $V_k$ .  $\square$

It follows readily that  $X_k$  is independent of  $\{V_k, V_{k-1}, \dots\}$ . Obviously  $X_k$  is dependent on  $A_k$ , but it is independent of  $\{A_{k-1}, A_{k-2}, \dots\}$ , since the  $A_k$ 's are independent. Since  $X_{k-l}$  is a function of  $A_{k-l}$  and  $V_{k-l}$ , it follows that  $X_k$  is independent of  $X_{k-l}$  for  $l \geq 1$ .

The approximation that the  $X_k$  are independent uniformly distributed random variables implies that the statistics of the precoded symbols are very similar to the statistics of the original data symbols, if the latter are independent identically distributed and are approximately equally likely to assume the  $M$  values in their alphabet. In particular, the variance of the precoded symbols,  $\text{Var}[X_k]$ , is approximately  $M^2/3$ , the variance of a uniform random variable. The original data symbols have variance  $(M^2 - 1)/3$ , if they are equally likely. This approximation suggests that for large  $M$ , the average power of the precoded symbols  $X_k$  is approximately the same as the average power of the original symbols. Thus, the modulo operation accomplishes the objective of reducing the peak and average transmitted power to approximately that of the original signal.

In the absence of channel coding, transmitter precoding does not offer a substantial advantage over the DFE, since all it accomplishes is to eliminate error propagation, which turns out to be a minor problem in practice. Transmitter precoding has the major disadvantage that it requires precise knowledge of the channel in the transmitter. For these reasons, it was not used until recently. However, it has recently become important as one of three available equalization methods for obtaining the best performance in combination with channel coding on channels with intersymbol interference (Chapter 14). The DFE, while widely used in uncoded systems, is fundamentally incompatible with channel coding because of the requirement for immediate symbol decisions to cancel postcursor ISI. Channel decoding inevitably introduces a multi-symbol delay in the detection and decision process.

## 10.2. GENERALIZED EQUALIZATION METHODS

In Section 10.1, two important equalization techniques were introduced, linear and decision-feedback equalization. The structures developed were optimal under the criterion of minimizing the noise at the slicer input subject to the constraint that there be no ISI. There are several directions that these results can be generalized, and all are important in practice:

- Remove the assumption that the front end of the receiver consist of a matched filter followed by a symbol-rate sampler. While this structure was shown to be optimal for Gaussian channel noise, it is problematic on many real channels because it requires knowledge of the channel transfer function and presumes that this transfer function is not changing with time. There are adaptive techniques for dealing with these problems (Chapter 11), but they typically work in discrete time. Thus, practical receivers for unknown or time-varying channels typically do not use front-end matched filtering, although as shown in Section 10.3 the equivalent operation can be performed in discrete time if the sampling rate is increased.
- Generalize the criterion of optimality to allow for residual ISI at the slicer. By allowing residual ISI, we can reduce the variance of the noise, and usually there is a net advantage in SNR at the slicer.

- Constrain the complexity of the equalization filters, so that we can make them practical to implement and control the implementation cost.

In this section, we will deal with two of these issues. First we will not assume a matched-filter receiver front end. Second, we will define a *mean-square error (MSE)* criterion that takes into account ISI as well as noise. We will design optimal equalizers under both the zero-forcing (ZF) and MSE criteria. In this section we will stick with symbol-rate sampling, relaxing that assumption in Section 10.3, and we will not constrain the equalizer complexity, leaving that consideration to Chapter 11 where adaptive equalization is covered.

### 10.2.1. Preliminaries

Before deriving optimal equalizer structures, we will choose a basic discrete-time channel model that does not presume front-end matched filtering, as in Section 10.1, and also define the MSE criterion.

#### Data-Symbol and Channel Model

The noise, signal, and channel model we consider here is shown in Figure 10-11a. Unlike for the ZF criterion, to design receivers according to the MSE criterion we must assume a statistical model for the data symbols. Hence, the lower case deterministic signals, like  $\{a_k\}$ , become upper case, like  $\{A_k\}$ , denoting a random process. The complex-valued data symbols  $A_k$  and additive complex noise are both assumed to be zero-mean wide-sense stationary discrete-time random processes with power spectra

$$S_A = A_a^2 \cdot G_a G_a^* \quad (10.48)$$

$$S_Z = A_z^2 \cdot G_z G_z^* \quad (10.49)$$

where the minimum-phase spectral factorizations of Section 2.5.2 have been introduced, and  $G_a$  and  $G_z$  are loosely minimum-phase, causal, monic transfer functions. In case these are white, the power spectra reduce to  $A_a^2$  or  $A_z^2$  respectively. We also make the reasonable assumption that the noise and data symbols are uncorrelated and independent. The channel  $H$ , presumed to be rational, introduces *dispersion* or *inter-symbol interference (ISI)*. It is no longer assumed that  $H$  is non-negative real on the unit circle, as it would be with a MF front end. For purposes of analysis, it is convenient to decompose the rational  $H$  canonically in the manner of (2.44),

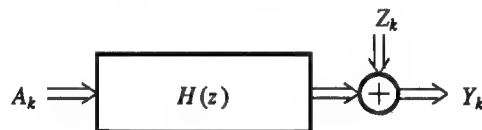


Figure 10-11. A discrete-time channel model sampled at the symbol rate.

$$H = H_0 \cdot z^r \cdot H_{\min} \cdot H_{\max} \cdot H_{\text{zero}} \quad (10.50)$$

where  $H_0$  is a complex constant,  $H_{\min}$  is monic, causal, and minimum-phase,  $H_{\max}$  is monic, anti-causal, and maximum-phase, and  $H_{\text{zero}}$  contains all the zeros on the unit circle, and is causal and monic. Generally a flat delay in the channel is of no consequence, so we will assume that  $r = 0$ .

### Example 10-9.

When the receiver front-end is a matched filter followed by symbol-rate sampling,  $H = S_h$ .

For this case,  $H_{\min} = H_{\max}^*$ ,  $H_0 = A_h^2$ , and  $H_{\text{zero}}$  assumes a particular form where zeros come in pairs.  $\square$

### Example 10-10.

Given the channel

$$H(z) = \frac{(1 - 0.1z^{-1})(1 - 2z^{-1})(1 - z^{-1})}{(1 - 0.5z^{-1})} \quad (10.51)$$

the minimum-phase and unit-circle zero terms are all in monic form, but a bit of manipulation is required on the maximum-phase term,

$$(1 - 2z^{-1}) = -2z^{-1}(1 - 0.5z) \quad (10.52)$$

and hence we can identify

$$H_0 = -2, \quad r = -1, \quad H_{\min}(z) = \frac{(1 - 0.1z^{-1})}{(1 - 0.5z^{-1})}, \quad (10.53)$$

$$H_{\max}(z) = (1 - 0.5z), \quad H_{\text{zero}}(z) = (1 - z^{-1}).$$

Then we can set  $r = 0$ , in effect dealing with the channel  $zH(z)$  rather than  $H(z)$ .  $\square$

## Physical Constraints on the Channel Model

Physical channels will never have a left-sided component to the impulse response. Thus we would expect  $H_{\max}$  to be an FIR filter, implying that it has poles only at  $z = \infty$ , because there are no poles in  $H$  outside the unit circle (except possibly at  $z = \infty$ ). Channel poles outside the unit circle, while mathematically feasible, can be ruled out by physical considerations.

### Example 10-11.

The equivalent channel response for the WMF is  $S_h$ , which is non-negative real on the unit circle. Excluding poles at  $z = 0$  and  $z = \infty$ , if  $S_h$  has any additional poles at all, then it will have poles outside the unit circle. The practical problem here is that if the received pulse  $h(t)$  is right sided but does not have finite support (it decays to zero but never reaches zero), the matched filter will be left-sided and will not have finite support, and hence is not physically realizable. Under this condition, the matched filter can only be approximated by a filter with a finite-support impulse response, and the resulting response will again be right-sided. For this practical approximation, unlike the ideal matched filter case,  $H$  can have poles inside the unit circle, but not outside.  $\square$

While we expect  $H_{\max}$  to be FIR, it is not necessarily unity. That is, channel zeros

outside the unit circle are feasible, and common if we use the WMF front end.

### Example 10-12.

The channel model for the mobile radio channel in Section 5.4 for the broadband case, where the delay spread is large, consists of independent Rayleigh fading channels with different delays (see Figure 5-33). Assume that there are two resolvable paths, so the impulse response of the channel is  $c_1 h(t) + c_2 h(t - \tau)$  where  $c_1$  and  $c_2$  are independent Gaussian random variables. (Actually,  $c_1$  and  $c_2$  are random processes, but we are interested in them at some fixed time.) The  $\tau$  is the differential delay of the two paths, and  $h(t)$  is the impulse response of the ideal channel, typically dominated by the transmit and receive filters, and chosen to satisfy the Nyquist criterion. If the output of the receive filter is sampled at the symbol rate, then the discrete-time channel has impulse response  $h_k = c_1 \delta_k + c_2 h(kT - \tau)$ . Just to illustrate what can happen, for simplicity assume that  $\tau$  is a multiple of the symbol interval  $T$ ,  $\tau = mT$ , so that

$$h_k = c_1 \delta_k + c_2 \delta_{k-m}, \quad H(z) = c_1 + c_2 z^{-m}. \quad (10.54)$$

This channel model has poles only at  $z = 0$ , and the  $m$  zeros satisfy

$$z^m = -\frac{c_2}{c_1}, \quad |z| = \left| \frac{c_2}{c_1} \right|^{1/m}. \quad (10.55)$$

Hence all  $m$  zeros have the same radius  $|c_2|/|c_1|$ . The channel is minimum-phase if and only if  $|c_2| \leq |c_1|$ . Thus, the channel will have zeros outside the unit circle, and not be minimum-phase, if the greater delay path has a higher strength than the lesser delay path. If the lesser delay path is the direct path to the transmitter, this is unlikely to happen. However, if both paths are reflected, as during a deep shadowing, then the channel could easily not be minimum-phase. Furthermore, due to the fading, the channel is likely to alternate between minimum and non-minimum phase.  $\square$

As we will see shortly, these zeros outside the unit circle, if they exist, are quite problematic for both the LE and DFE, because they require equalization filters with poles outside the unit circle. Practically speaking these zeros outside the unit circle can only be approximately equalized, and accurate equalization requires high-complexity filters. The situation described in Example 10-12 is particularly difficult for equalization, because the channel can be minimum- and non-minimum-phase at different times. As we will see, the desired structure for the equalizer when the channel model has zeros outside the unit circle is much different than when it is minimum-phase.

### Mean-Square Error

In this section we want to allow residual ISI at the slicer input, in which case there is not only noise but ISI at the slicer input. The simple  $Q(\cdot)$  formula for the probability of error no longer applies. Rather than calculate the exact probability of error, which is difficult, we will compare and optimize equalizers using an MSE criterion. The MSE is simply the variance of the error between the slicer input and the actual data symbol. We denote it by  $\epsilon^2$ , and define it as

$$\epsilon^2 = E[|E_k|^2], \quad E_k = Q_k - A_k, \quad (10.56)$$

where  $A_k$  is the data symbol, assumed to be a random variable, and  $Q_k$  is the input sample to the slicer. The expectation is with respect to both the data-symbol statistics

and the noise statistics. To calculate the MSE, it is necessary to know the power spectrum of the additive channel noise,  $S_Z$ , the power spectrum of the data symbols,  $S_A$  (assuming the data symbols are modeled as a wide-sense stationary random process), and the equivalent channel response  $H$ .

In Section 10.1 we used a "figure of merit"  $\gamma$  to compare different equalization techniques, where the probability of error was directly related to  $\gamma$  through the  $Q(\cdot)$  function. It is natural to define a quantity similar to the figure of merit, where we replace the Gaussian noise variance in the denominator by half the MSE (half because the MSE is the variance of the complex-valued error, and  $\sigma^2$  is the variance of only the real or imaginary component),

$$\gamma = \frac{a_{\min}^2}{\epsilon^2/2}. \quad (10.57)$$

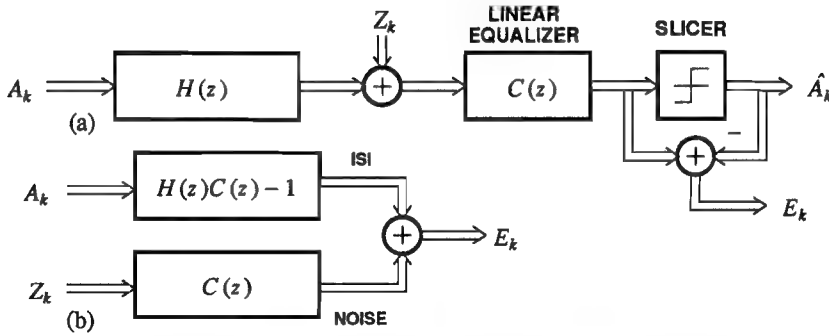
The philosophy here is that the residual ISI is a source of "noise" similar to the additive Gaussian noise, so we simply add its variance. In fact, on some channels the ISI might be approximately Gaussian, by the central limit theorem, and then a good approximation to the error probability would be  $K \cdot Q(\frac{1}{2}\sqrt{\gamma})$ . Even without having to make this approximation, minimizing  $\epsilon^2$  is a reasonable criterion for the design of the equalizer.

In this section we will simply use  $\epsilon^2$  as a measure of the effect of ISI and noise, and design and compare different receiver structures based on this measure. Since  $E_k$  is not Gaussian, this approach is not equivalent to minimizing the error probability, but because of its simplicity, it is widely applied in practice. As we will see in Chapter 11, this criterion is also widely used for the adaptation of equalizers as well.

### 10.2.2. Linear Equalizer

The linear equalizer (LE) applies the channel output to an equalizer filter  $C$ , as shown in Figure 10-12, the purpose of which is to reduce or eliminate the ISI. The error  $E_k$  is the difference between the slicer input and the current data symbol, which we want to be as small as possible. As is conventional, the error is shown as the difference between the slicer input and output, called the slicer error, which will be the same as  $E_k$  as defined in (10.56) only if the slicer makes a correct decision. In this section we will assume that the slicer always makes a correct decision for purposes of analysis of the MSE and design of the equalizer filters (this assumption becomes relevant only in the case of the DFE). With this assumption, the slicer error can be generated in the convenient form of Figure 10-12a.

Assuming that the slicer decisions are correct,  $A_k = \hat{A}_k$ , then the filters in Figure 10-12a can be combined as shown in Figure 10-12b. This illustrates clearly that  $E_k$  consists of two components: the output of the top filter is the residual ISI after equalization, and the output of the bottom filter is the noise component at the slicer input. There is generally a tradeoff between these two components of error. Minimizing the ISI enhances the noise, so if we are willing to accept more ISI after equalization we can reduce the noise enhancement. Designing  $C$  in accordance with the MSE criterion represents one way to specify the desired tradeoff between noise enhancement



**Figure 10-12.** A linear equalizer receiver. (a) The receiver uses an equalizer filter  $C$  and slicer, generating error  $E_k$ . (b) Equivalent way of generating the error assuming that decisions are correct.

and ISI.

The power spectrum of the slicer error evaluated on the unit circle, from Figure 10-12b and the assumption of independence between the data symbols and noise, is

$$S_E = S_A \cdot |HC - 1|^2 + S_Z \cdot |C|^2. \quad (10.58)$$

The *mean-square error (MSE)*, which is the variance of the slicer error,  $E[|E_k|^2]$ , is the integral of the power spectrum given by (10.58),

$$\epsilon^2 = \langle S_E \rangle_A. \quad (10.59)$$

We will now determine the equalizer filter  $C$  for two cases: First, we will constrain the ISI to be zero (the ZF criterion) and then we will not constrain the ISI (the MSE criterion).

### Zero-Forcing Criterion

In Section 10.1 we constrained the ISI to be zero at the slicer. We now repeat that design for a general channel model  $H$ . The equalizer is simply chosen to force the ISI component of the slicer error to zero (hence the name zero-forcing), or

$$C = \frac{1}{H} = \frac{1}{H_0 \cdot H_{\min} \cdot H_{\max} \cdot H_{\text{zero}}}. \quad (10.60)$$

We can make several observations:

- $C$  does not depend on either the data-symbol or noise statistics. The MSE will depend on the noise spectrum, although not on the data-symbol spectrum because the ISI is forced to zero.
- This equalizer cannot be stable unless  $H_{\text{zero}} = 1$ ; that is, the channel has no zeros on the unit circle.
- $H_{\min}^{-1}$  is a readily implemented causal minimum-phase filter.



- $H_{\max}^{-1}$  is an anti-causal maximum-phase filter. In the practical case where  $H_{\max}$  is an FIR filter (since poles would result in an impractical left-sided channel impulse response),  $H_{\max}^{-1}$  is an all-pole IIR anticausal filter, which is impractical to implement and can at best be approximated.

To summarize, a minimum-phase channel is relatively straightforward to equalize. When the channel has a maximum-phase component, even if that component is FIR, the LE-ZF suddenly becomes impractical to implement and can at best be approximated. This approximation can be an FIR filter, but will often require a high order to be accurate.

The only component of slicer error is the noise, which has, from (10.58) and (10.60), variance

$$\epsilon_{\text{LE-ZF}}^2 = \langle S_Z / |H|^2 \rangle_A. \quad (10.61)$$

For white noise and rational  $H$ , a convenient way to evaluate  $\epsilon_{\text{LE-ZF}}^2$  without the need to evaluate an integral is to note that it is  $A_z^2$  times the coefficient of  $z^0$  in the expansion of  $(HH^*)^{-1}$ .

#### Example 10-13.

For white noise ( $S_Z(z) = A_z^2$ ) and a first-order FIR channel,  $H(z) = 1 - cz^{-1}$  for some complex-valued  $c$ , the LE-ZF is  $C(z) = 1/(1 - cz^{-1})$ . When the channel is minimum-phase ( $|c| < 1$ ) the equalizer impulse response is  $c_k = c^k u_k$ , where  $u_k$  is the unit step function, a stable causal response. When the channel is not minimum-phase ( $|c| > 1$ ), the impulse response is anticausal and IIR,  $c_k = -c^k u_{-k-1}$ . The noise is processed only by  $C(z)$ , and hence has power spectrum  $A_z^2 C(z) C^*(1/z^*)$ . The MSE can be evaluated from the partial fraction expansion

$$C(z)C^*(1/z^*) = \frac{1}{H(z)H^*(1/z^*)} = \frac{1}{1 - |c|^2} \left[ \frac{c z^{-1}}{1 - c z^{-1}} + \frac{1}{1 - c^* z} \right]. \quad (10.62)$$

For the minimum-phase case, the first term in (10.62) starts at  $k = 1$ , and does not contribute to the  $z^0$  term. The second term does, and  $\epsilon_{\text{LE-ZF}}^2 = A_z^2/(1 - |c|^2)$ . For the non-minimum-phase case, rewrite (10.62) as

$$C(z)C^*(1/z^*) = \frac{1}{H(z)H^*(1/z^*)} = \frac{1}{|c|^2 - 1} \left[ \frac{1}{1 - c^{-1}z} + \frac{c^* z^{-1}}{1 - c^* z^{-1}} \right]. \quad (10.63)$$

In this case, only the first term contributes to  $z^0$ , and  $\epsilon_{\text{LE-ZF}}^2 = A_z^2/(|c|^2 - 1)$ . In both cases  $\epsilon_{\text{LE-ZF}}^2 \rightarrow \infty$  as  $|c| \rightarrow 1$ . As noted in Section 10.1.2, the LE-ZF is not useful when the channel model has zeros on the unit circle.  $\square$

#### Example 10-14.

We can now rederive the results of Section 10.1 by setting  $H = G_h$  and  $S_Z = 2N_0/A_h^2$ , where the spectral factorization of  $S_h$  is  $S_h = A_h^2 G_h G_h^*$ . This corresponds to the symbol-rate discrete-time channel model for the WMF front end. In this case, the LE-ZF is  $C = G_h^{-1}$  and the MSE is

$$\epsilon_{\text{LE-ZF}}^2 = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{2N_0 A_h^2}{|G_h|^2} d\omega = 2N_0 \cdot \langle S_h^{-1} \rangle_A. \quad (10.64)$$

Equation (10.64) is consistent with (10.31). For this case, it *appears* that the equalizer is relatively simple to implement because the channel model is strictly minimum-phase. However, that is *not* the case, because the trouble is hidden in the WMF, which includes a strictly maximum-phase filter  $1/G_h^*$ . That WMF filter can only be approximated in practice if  $G_h$  has any zeros, since the WMF will then have poles outside the unit circle. The continuous-time matched filter may also be problematic if the received pulse  $h(t)$  is causal and has unbounded support.  $\square$

## Mean-Square Error Criterion

The second method for designing the equalizer is to minimize the mean-square error (MSE)  $E[|E_k|^2]$ , taking into account both the ISI and noise components. The resulting equalizer is known as the mean-square error linear equalizer (LE-MSE). Since  $C$  is not constrained in complexity (this assumption will be removed in Chapter 11),  $C$  can be chosen independently at each frequency. The MSE can therefore be minimized by minimizing  $S_E$  given by (10.58) at each frequency by judicious choice of  $C$ .

First note that the power spectrum of the signal plus noise at the channel output in Figure 10-11 is

$$S_Y = S_A \cdot |H|^2 + S_Z. \quad (10.65)$$

It will turn out that this power spectrum plays a crucial role in the equalizer design.

### Exercise 10-2.

Show that by completing the square, (10.58) can be written as

$$S_E = S_Y \cdot |C - S_A S_Y^{-1} H^*|^2 + S_A S_Z S_Y^{-1}. \quad (10.66)$$

$\square$

Since all terms in (10.66) are positive, it can be minimized at each frequency by forcing the first term to zero,

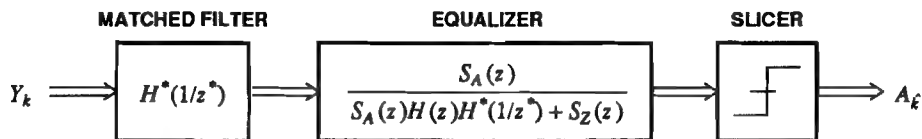


Figure 10-13. The LE-MSE receiver, which minimizes the mean-square slicer error, consists of a MF and equalizer.

$$C = S_A S_Y^{-1} H^* \quad (10.67)$$

which is shown in Figure 10-13. We recognize that the term  $H^*$  is a discrete-time MF, and separate it out. Again, we can make some important observations:

- The reason that the LE-ZF does not contain a MF is that the equalizer would simply find the inverse of this filter, making it pointless. However, the LE-MSE includes a discrete-time matched filter.
- The MF is typically difficult to realize when the channel response has poles, as noted previously. Those poles will, for a physically meaningful channel and continuous-time receive filter, be inside the unit circle, which will place poles outside the unit circle in the MF. The resulting anticausal IIR filter can only be approximated by a realizable filter.
- The equalizer  $S_A S_Y^{-1}$ , without the MF included, is non-negative real on the unit circle. Excluding poles at  $z = 0$  and  $z = \infty$ , if it has any additional poles, then some of these must be outside the unit circle, making it unrealizable.
- If, as is typically the case,  $S_Z \neq 0$  on the unit circle (the channel noise is non-zero at all frequencies), then the equalizer  $S_A S_Y^{-1}$  can have no poles on the unit circle, and hence is stable. Similarly, since the channel  $H$  is assumed to be stable and have no poles on the unit circle, then the matched filter  $H^*$  will also be stable. Hence, for this case the LE-MSE is stable, in contrast to the LE-ZF, which will have poles on the unit circle if  $H$  has zeros on the unit circle. However, it is possible for the LE-MSE to have poles outside the unit circle as noted above, in which case it can only be approximated in practice even though it is mathematically well defined as a stable filter.
- As  $S_Z \rightarrow 0$ , the LE-MSE approaches the LE-ZF, which is expected since in that case the LE-MSE will ignore the noise and focus on minimizing the ISI.

To summarize, the practicality of the LE-MSE is quite distinct from the LE-ZF. The LE-ZF has difficulty with non-minimum-phase channels, in the sense that the equalizer can only be approximated, and that approximation may have relatively high complexity. The LE-MSE has a similar difficulty with any channel with poles, except at  $z = 0$  and  $z = \infty$ , due to the MF portion. The LE-ZF is not stable for channels with zeros on the unit circle. On the other hand, the LE-MSE generally has no difficulty with non-minimum-phase channels, where there are zeros outside the unit circle, or with channels with zeros on the unit circle.

Assuming (10.67), the power spectrum of the slicer error is the last term in (10.66), and hence the MSE is

$$\epsilon_{\text{LE-MSE}}^2 = \langle S_Z / (|H|^2 + S_Z S_A^{-1}) \rangle_A. \quad (10.68)$$

Comparing with (10.61), the extra  $S_Z S_A^{-1}$  term in the denominator ensures that  $\epsilon_{\text{LE-MSE}}^2 \leq \epsilon_{\text{LE-ZF}}^2$ , since the integrand must be smaller at some frequencies in the MSE case. Furthermore, if  $S_Y S_A^{-1} \neq 0$  on the unit circle, which will typically be the case, then  $\epsilon_{\text{LE-MSE}}^2$  is guaranteed to be finite regardless of the channel  $H$ . Thus the LE-MSE is guaranteed to be stable, although it may not have a right-sided impulse response. This nice property follows intuitively from the fact that this equalizer can

avoid infinite noise enhancement (which plagues the LE-ZF) by allowing some residual ISI at the slicer.

### Example 10-15.

Continuing Example 10-14, where the receiver front end is a WMF, assume that the data symbols are white ( $S_A = A_d^2$ ). Then the equalizer is

$$C = \frac{S_A G_h^*}{S_A |G_h|^2 + S_Z} = \frac{G_h^*}{|G_h|^2 + 2N_0/A_h^2 A_d^2}, \quad (10.69)$$

and we see that  $C \rightarrow G_h^{-1}$  (the LE-ZF solution) as  $2N_0/A_h^2 A_d^2 \rightarrow 0$  (the high-SNR case). The MSE is given in integral form by

$$\epsilon_{\text{LE-MSE}}^2 = \langle 2N_0 / (S_h + 2N_0/A_d^2) \rangle_A, \quad (10.70)$$

which approaches the ZF-LE as  $N_0 \rightarrow 0$ .  $\square$

## 10.2.3. Decision-Feedback Equalizer

To determine the optimal DFE under an MSE criterion, we will draw heavily on the connection between the DFE and linear prediction established in Section 10.1.3. The opportunity for improving on the MSE of the LE comes from the fact that the slicer error samples are correlated, and hence can be reduced by a linear prediction error filter. The LE slicer error  $E_k$  from Figure 10-12b is reproduced in Figure 10-14a, with the addition of a linear prediction filter  $E$ , which is constrained to be causal and monic. We know that the new slicer error  $E_k'$  will have a smaller variance than the LE slicer error  $E_k$  if  $E$  is chosen properly, but the question is whether the slicer error configuration of Figure 10-14a corresponds to a practical DFE configuration. Fortunately, the DFE configuration of Figure 10-14b is equivalent, assuming that decisions are correct ( $\hat{A}_k = A_k$ ). Furthermore, the feedback filter is realizable, because the postcursor equalizer in the feedback loop,  $(E - 1)$ , is a strictly causal filter; that is, the zero-delay coefficient is zero since  $E$  is monic. Thus, the output of this filter, subtracted from the slicer input, is a function of *past* decisions only, as it must be to avoid zero delay in the feedback loop.

We know from the properties of optimal prediction in Section 3.2.3 that if the prediction error filter is properly chosen, then the slicer error  $E_k'$  in Figure 10-14 will have a variance no larger than  $E_k$ , the LE slicer error, for the same choice of  $C$ . Furthermore, we know that the DFE slicer error must be white for a properly chosen prediction error filter  $E$ . Thus, the addition of the feedback filter in Figure 10-14 must be beneficial, in the sense of reducing the mean-square slicer error, when compared to a LE using the same  $C$ .

Having defined the DFE structure, it remains to determine the optimal equalizers (precursor equalizer  $CE$  and postcursor equalizer  $E - 1$ ). It is much simpler to determine the optimal  $C$  and  $E$ , and then infer the precursor and postcursor equalizers. In fact, for any  $C$ ,  $E$  is chosen as the optimal linear prediction error filter. Once  $E$  is so determined, it is simple to determine the optimal  $C$ .

As in the case of the LE, there are two alternative approaches: the zero-forcing DFE (DFE-ZF) forces the ISI to zero at the slicer input, while the mean-square DFE (DFE-MSE) minimizes the variance of the slicer error. It turns out to be easy to show that  $C$  is the same for the LE-ZF and DFE-ZF, and likewise for the LE-MSE and DFE-MSE. Thus, for either criterion the only difference between the LE and DFE is the addition of the linear prediction error filter in the DFE. In both cases, the linear prediction filter results in a white slicer error process, although in the ZF case that process is Gaussian and in the MSE case it is not (because of the residual ISI).

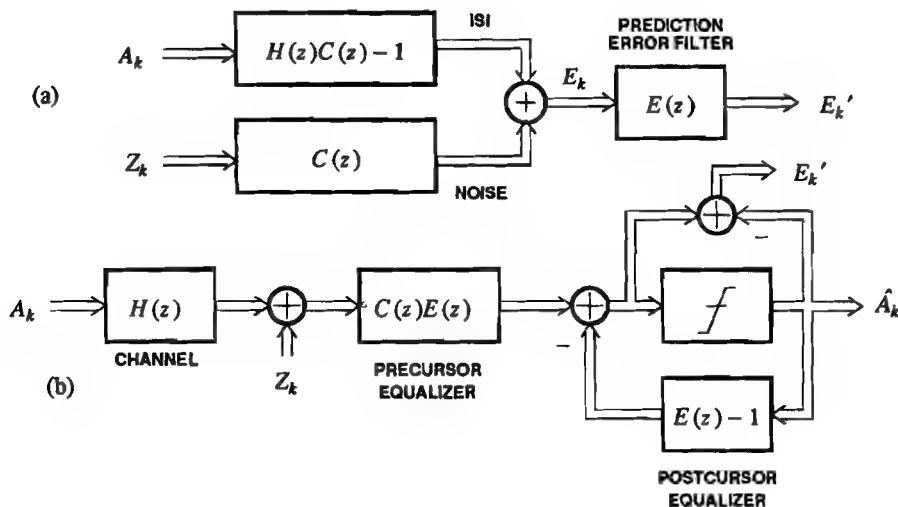
Let us first assume some filter  $C$ , and choose the optimal linear predictor  $E$ . For a given  $C$ , the power spectrum of the LE slicer error is given by (10.58), and a monic minimum-phase spectral factorization of this spectrum can be performed,

$$S_E = \epsilon_{\text{DFE}}^2 G_e G_e^* \quad (10.71)$$

where  $\epsilon_{\text{DFE}}^2$  is the variance of the innovations  $E_k'$ , and hence is the mean-square DFE slicer error. The prediction error filter is the monic minimum-phase whitening filter  $E = G_e^{-1}$ . A convenient formula for  $\epsilon_{\text{DFE}}^2$  is given by (2.57),

$$\epsilon_{\text{DFE}}^2 = \langle S_E \rangle_G. \quad (10.72)$$

Naturally,  $S_E$ ,  $G_e$ , and  $E$  all depend on  $C$ , which depends in turn on the criterion used.



**Figure 10-14.** The decision-feedback equalizer (DFE). (a) Adding a linear prediction filter to Figure 10-12b to whiten the slicer error. (b) An equivalent DFE structure assuming that slicer decisions are correct.

### Zero-Forcing Criterion

Considering first the DFE-ZF, assume that there are no zeros on the unit circle ( $H_{\text{zero}} = 1$ ) so that the LE-ZF is stable (we will relax this assumption momentarily). The LE-ZF is then  $C = H^{-1}$ , and the only component of the LE slicer error is the noise passing through  $C$ . The slicer error spectrum is

$$S_E = \frac{S_Z}{HH^*} = \frac{A_z^2 G_z G_z^*}{|H_0|^2 \cdot H_{\min} H_{\max} H_{\min}^* H_{\max}^*} \quad (10.73)$$

and we can immediately see that  $\epsilon_{\text{DFE-ZF}}^2 = A_z^2 / |H_0|^2$ . A more universal formula follows from (10.72),

$$\epsilon_{\text{DFE-ZF}}^2 = \langle S_Z / |H|^2 \rangle_G. \quad (10.74)$$

Furthermore, the prediction error filter (monic minimum-phase whitening filter) is

$$E = \frac{H_{\min} H_{\max}^*}{G_z}, \quad (10.75)$$

thereby determining the precursor equalizer,

$$CE = H^{-1} \cdot \frac{H_{\min} H_{\max}^*}{G_z} = \frac{1}{H_0} \cdot \frac{H_{\max}^*}{H_{\max}} \cdot G_z^{-1}. \quad (10.76)$$

A few observations:

- The term  $G_z^{-1}$  is a minimum-phase noise-whitening filter. This ensures that the noise at the slicer input is white. This is expected, since the ISI is completely eliminated by the precursor and postcursor equalizers, and thus the noise is the only component of slicer error. The slicer error will always be white for an optimal DFE.
- The  $H_{\max}^* / H_{\max}$  term is an allpass filter. With a phase-only filtering, the precursor ISI can be eliminated, and there is no noise enhancement due to frequency-dependent precursor equalizer gain. In contrast, the LE eliminates *both* precursor and postcursor ISI at the expense of noise enhancement.
- The allpass filter does not modify the noise spectrum at the slicer input, allowing the noise-whitening filter to do its job without interference.
- The response of the channel plus precursor equalizer is  $E$ , which is minimum-phase. The role of the allpass filter is thus to convert the maximum-phase channel component to minimum-phase by reflecting poles and zeros inside the unit circle. Another interpretation of this result is that among all causal responses with the same magnitude Fourier transform, the minimum-phase response has maximum energy near  $k = 0$  (Problem 2-22). Thus, in this sense the minimum-phase response minimizes the energy of the ISI which must be canceled by the postcursor equalizer, minimizing the signal energy that is thrown away by canceling it in the postcursor equalizer.
- If the channel is minimum-phase and the noise is white, no precursor equalizer at all is required, aside from a flat gain! Thus, the DFE-ZF is like the LE-ZF, in

that it finds minimum-phase channels much easier to deal with. In fact, except for the noise-whitening filter required when the noise is not white, the precursor equalizer is unnecessary!

- If there is a maximum-phase component  $H_{\max}$ , then in a practical sense it must be an FIR filter (all its poles are at  $z = \infty$ ). In this case the allpass filter component of the precursor equalizer has poles outside the unit circle, and hence can only be approximated by a (relatively high-complexity) FIR filter. Thus, the DFE-ZF has a similar difficulty as the LE-ZF with non-minimum-phase channels.

#### Example 10-16.

Consider the same channel as in Example 10-13,  $H(z) = 1 - cz^{-1}$ . For the minimum-phase case,  $|c| < 1$ . We identify  $H_0 = 1$  and  $H_{\max}(z) = 1$ , and hence no precursor equalizer is needed. Because there is no precursor equalizer, the slicer error is  $\epsilon_{\text{DFE-ZF}}^2 = A_z^2$ , and there is no noise enhancement. The postcursor equalizer,  $E(z) - 1 = -cz^{-1}$ , simply cancels the single ISI sample. Note that nothing special happens as  $|c| \rightarrow 1$ , in sharp contrast to the LE-ZF, which is not stable in that case.  $\square$

#### Example 10-17.

Repeating Example 10-16 for the non-minimum-phase case,  $|c| > 1$ , we will obtain a much different answer! Rewriting  $H(z) = -cz^{-1}(1 - c^{-1}z)$  we can ignore the  $z^{-1}$  term, and identify  $H_0 = -c$  and  $H_{\max}(z) = 1 - c^{-1}z$ . The MSE is thus  $\epsilon_{\text{DFE-ZF}}^2 = A_z^2/|c|^2$ , and thus the slicer error is smaller than in the minimum-phase case (since the receiver is basing its decision on the larger delayed sample  $c$ ). Since  $E(z) = 1 - (c^*)^{-1}z^{-1}$ , the postcursor equalizer is  $E(z) - 1 = -(c^*)^{-1}z^{-1}$ , a single tap as in the minimum-phase case. The big difference is in the precursor equalizer, which is the allpass filter

$$C(z)E(z) = -\frac{1}{c} \cdot \frac{1 - (c^*)^{-1}z^{-1}}{1 - c^{-1}z} \quad (10.77)$$

This precursor equalizer is anticausal, since it has a pole outside the unit circle, and can at best only be approximated by a realizable filter. There is thus a wide gap between the complexity of the precursor equalizer for the minimum-phase and non-minimum-phase channels. When  $|c|$  crosses unity, in principle nothing bad happens (in contrast to the LE-ZF), but in practice the structure of the precursor equalizer changes dramatically. This can present implementation difficulties; and is a real problem for example on broadband Rayleigh fading channels (Example 10-12).  $\square$

It should be noted from these two examples that the minimum-phase solution will work in the maximum-phase case, in the sense of eliminating the ISI component of the slicer error, but the penalty paid will be a larger MSE slicer error ( $A_z^2$  rather than  $A_z^2/|c|^2$ ). This statement is more generally true: as long as the maximum-phase channel component has poles only at  $z = 0$  or  $z = \infty$ , which is normally expected on physical grounds, zero ISI at the slicer can be ensured by a postcursor equalizer only (if an appropriate delay is added to the channel) because the channel has a right-sided impulse response. However, a penalty in MSE is paid for not equalizing to a minimum-phase response.

**Example 10-18.**

Continuing Example 10-14, assume that the front end is a WMF. Then  $C = 1/G_h$ , and the power spectrum of the LE slicer error is

$$S_E = \frac{2N_0/A_h^2}{|G_h|^2} = \frac{2N_0/A_h^2}{G_h G_h^*}. \quad (10.78)$$

The optimal predictor is thus the inverse of the minimum-phase portion,  $E = G_h$ , and the forward equalizer is  $CE = 1$ ; that is, as expected, the precursor equalizer is actually the WMF. The MSE is given by

$$\epsilon_{DFE-ZF}^2 = \frac{2N_0}{A_h^2} = 2N_0 \cdot \langle S_h^{-1} \rangle_G, \quad (10.79)$$

consistent with (10.39).  $\square$

One of the practically important properties of the DFE-ZF is that, in contrast to the LE-ZF, it works perfectly well in the presence of zeros on the unit circle. To see this, assume that  $H_{\text{zero}} \neq 1$ , but design the precursor equalizer as before, turning the maximum-phase component into a minimum-phase component. The response of channel plus precursor equalizer is then  $EH_{\text{zero}}$  rather than  $E$ . Since this response is still causal and monic, the ISI can still be canceled by a strictly causal postcursor equalizer  $EH_{\text{zero}} - 1$ . In effect we have included  $H_{\text{zero}}$  in the minimum-phase component of the channel, which is customary and introduces no mathematical difficulties.

**Example 10-19.**

Replace  $H(z)$  in Example 10-17 by  $H(z) = (1 - cz^{-1})(1 - z^{-1})$  for  $|c| > 1$ . Retaining the same allpass precursor equalizer, the isolated pulse response at the precursor equalizer output is

$$(1 - (c^*)^{-1})(1 - z^{-1}) = 1 - (c^* + 1)(c^*)^{-1}z^{-1} + (c^*)^{-1}z^{-2} \quad (10.80)$$

and the ISI can be canceled with postcursor equalizer  $-(c^* + 1)(c^*)^{-1}z^{-1} + (c^*)^{-1}z^{-2}$ .  $\square$

**Mean-Square Error Criterion**

The DFE-MSE optimal filters are almost as easy to determine. The first observation is that (10.72) is monotonically increasing in  $S_E$  at each frequency, and hence will be minimized by choosing  $C$  at each frequency to minimize  $S_E$ . Thus  $C$  for the DFE-MSE is precisely the same as  $C$  for the LE-MSE, since  $C$  was designed to meet the same criterion.  $C$  is given by (10.67) and  $S_E$  is given by the last term in (10.66). To find the whitening filter  $E$ , first do a minimum-phase spectral factorization of  $S_Y$ ,

$$S_Y = S_A \cdot HH^* + S_Z = A_y^2 \cdot G_y G_y^*, \quad (10.81)$$

$$A_y^2 = \langle S_A |H|^2 + S_Z \rangle_G, \quad (10.82)$$

and thus from (10.66)



$$S_E = \frac{S_A S_Z}{S_Y} = \frac{A_a^2 A_z^2}{A_y^2} \cdot \frac{G_a G_a^* G_z G_z^*}{G_y G_y^*} = \epsilon_{\text{DFE-MSE}}^2 G_e G_e^* . \quad (10.83)$$

It follows that

$$\epsilon_{\text{DFE-MSE}}^2 = \frac{A_a^2 A_z^2}{A_y^2} = \langle S_Z / (|H|^2 + S_Z S_A^{-1}) \rangle_G , \quad (10.84)$$

and

$$E = \frac{1}{G_e} = \frac{G_y}{G_a G_z} . \quad (10.85)$$

The optimal precursor equalizer is

$$CE = \frac{A_a^2}{A_y^2} \cdot H^* \cdot \frac{G_a^*}{G_y^*} \cdot G_z^{-1} . \quad (10.86)$$

As in the LE-MSE, this solution includes a matched filter  $H^*$ , and as in the DFE-ZF it includes a noise-whitening filter  $G_z^{-1}$ . The remaining term is, in contrast to the DFE-ZF, not an allpass filter. It is straightforward to verify that this MSE solution approaches the ZF solution as  $S_Z \rightarrow 0$  (Problem 10-8), and it follows from (10.74) and (10.84) that  $\epsilon_{\text{DFE-MSE}}^2 \leq \epsilon_{\text{DFE-ZF}}^2$  because the integrand is smaller at some frequencies for the MSE criterion.

#### Example 10-20.

Continuing Example 10-14, assume that the front end is the WMF. The power spectrum of the equivalent channel output is

$$S_y = S_A |G_h|^2 + \frac{2N_0}{A_h^2} = A_y^2 G_y G_y^* . \quad (10.87)$$

The only simplification is that the noise-whitening filter is not required, since the noise at the WMF output is already white ( $G_z = 1$ ), and hence the postcursor equalizer is  $E = G_y / G_a$  and the precursor equalizer is

$$CE = \frac{A_a^2}{A_y^2} G_h^* \frac{G_a^*}{G_y^*} . \quad (10.88)$$

The MSE is

$$\epsilon_{\text{DFE-MSE}}^2 = \langle 2N_0 / (S_h + 2N_0 S_A^{-1}) \rangle_G . \quad (10.89)$$

□

### Unbiased Mean-Square Error

If we calculate the transfer function from the data symbol  $A_k$  to the slicer error for the DFE-MSE design, it is, from Figure 10-14,

$$(HC - 1)E = -\frac{A_z^2}{A_y^2} \cdot \frac{G_z^*}{G_a G_y^*}. \quad (10.90)$$

The striking property of this solution is that, since  $G_z^*/G_a G_y^*$  is monic, the slicer error has a component proportional to the current data symbol  $A_k$ , namely  $(-A_z^2/A_y^2) \cdot A_k$ . Thus, the total component of the current symbol at the slicer input is this error plus  $A_k$ , or

$$\frac{A_y^2 - A_z^2}{A_y^2} \cdot A_k. \quad (10.91)$$

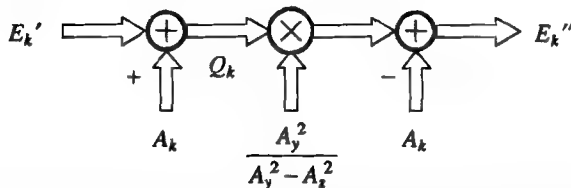
While this minimizes the MSE, the error probability can be made smaller by removing this bias in the amplitude of the data symbol reaching the slicer. (Recall that the slicer error is not Gaussian, so minimizing the MSE is not precisely the same as minimizing the error probability.) Thus, the performance can be improved by scaling the slicer input by a factor which removes this bias, this factor being  $A_y^2/(A_y^2 - A_z^2)$ . No other modifications are necessary, as the precursor and postcursor equalizers remain the same (since the gain is added after ISI cancellation). The resulting design is called the *unbiased DFE-MSE (DFE-MSE-U)* [9].

The MSE of the DFE-MSE-U,  $\epsilon_{\text{DFE-MSE-U}}^2$ , must be larger than  $\epsilon_{\text{DFE-MSE}}^2$ . We can find out what it is from Figure 10-15, which shows the relationship between the slicer error of the DFE-MSE,  $E_k'$ , and the DFE-MSE-U,  $E_k''$ . First, the DFE-MSE slicer input sample  $Q_k$  is obtained by adding the data symbol to  $E_k'$ , and then the slicer input is multiplied by the adjustment factor, and then the new slicer error is obtained by subtracting the data symbol. From this figure,

$$(A_y^2 - A_z^2) \cdot E_k'' = A_y^2 \cdot E_k' + A_z^2 \cdot A_k. \quad (10.92)$$

Finding the MSE from this relationship is easy for the particular case where the data symbols are zero-mean and independent ( $G_a = 1$ ). For this case,  $A_k$  is independent of  $E_k''$ , because the latter has no component of  $A_k$ , and  $A_k$  is independent of all the other components that make up  $E_k''$  (the noise and the other data symbols). Thus, we can easily calculate the variance,

$$A_y^4 \cdot \epsilon_{\text{DFE-MSE}}^2 = (A_y^2 - A_z^2)^2 \cdot \epsilon_{\text{DFE-MSE-U}}^2 + A_z^4 \cdot A_a^2, \quad (10.93)$$



**Figure 10-15.**  $E_k''$ , the slicer error for the unbiased DFE-MSE (DFE-MSE-U) obtained from  $E_k'$ , the slicer error for the DFE-MSE.

which, with the aid of (10.84) is easily solved for  $\epsilon_{\text{DFE-MSE-U}}^2$ ,

$$\epsilon_{\text{DFE-MSE-U}}^2 = \frac{A_a^2 A_z^2}{A_y^2 - A_z^2}. \quad (10.94)$$

The increase in MSE due to removing the bias is a factor of  $A_y^2/(A_y^2 - A_z^2)$ . Furthermore, it is readily shown that

$$\frac{A_a^2}{\epsilon_{\text{DFE-MSE}}^2} = \frac{A_a^2}{\epsilon_{\text{DFE-MSE-U}}^2} + 1, \quad (10.95)$$

and since the two ratios are in the form of signal-to-noise ratios (SNR's), we see that the SNR of the DFE-MSE-U is indeed smaller than the SNR of the DFE-MSE, in fact by exactly unity.

The DFE-MSE-U has two analytical disadvantages: the MSE is larger than for the DFE-MSE, and unlike the DFE-MSE the slicer error is not white. However, we do expect it to have a lower error probability. From a theoretical point of view, the DFE-MSE-U is related in a remarkable canonical way to the capacity of the discrete-time channel, as will be shown in Section 10.5.

It has also been shown [9] that among all DFE equalizers that are unbiased, the DFE-MSE-U achieves the minimum MSE. Since the DFE-ZF is also an unbiased DFE equalizer, it follows that

$$\epsilon_{\text{DFE-MSE}}^2 \leq \epsilon_{\text{DFE-MSE-U}}^2 \leq \epsilon_{\text{DFE-ZF}}^2. \quad (10.96)$$

#### 10.2.4. Maximum-Likelihood Sequence Detector

We demonstrated in Section 10.1 that the MLSD and the DFE-ZF share the same WMF front-end filtering. In Section 10.2.3 we generalized the DFE-ZF to channels that do not necessarily have a matched-filter front end. The Viterbi algorithm can be applied at the output of the DFE-ZF precursor equalizer, in place of the WMF. This is illustrated in Figure 10-16. Figure 10-16a shows a configuration in which the Viterbi algorithm is applied at the output of the precursor equalizer, and Figure 10-16b shows the equivalent channel model to the input of the Viterbi algorithm. This equivalent channel model displays all the characteristics needed to apply the Viterbi algorithm; the equivalent channel is causal and monic and the noise samples are Gaussian and independent.

The only restriction in using the Viterbi algorithm in detecting the data symbol sequence  $a_k$  under an ML criterion is that  $E$  be an FIR filter, so that the channel model is a finite-state machine. This requires the following conditions:

- $H_{\text{max}}^*$  must be an FIR filter, implying that, excluding poles at  $z = 0$  and  $z = \infty$ ,  $H_{\text{max}}$  have no poles. As we have discussed, this requirement is also necessary for physical realizability of the channel model.
- $H_{\text{min}}$  must be an FIR filter, implying that, excluding poles at  $z = 0$  and  $z = \infty$ , the channel model  $H$  must have no poles at all.

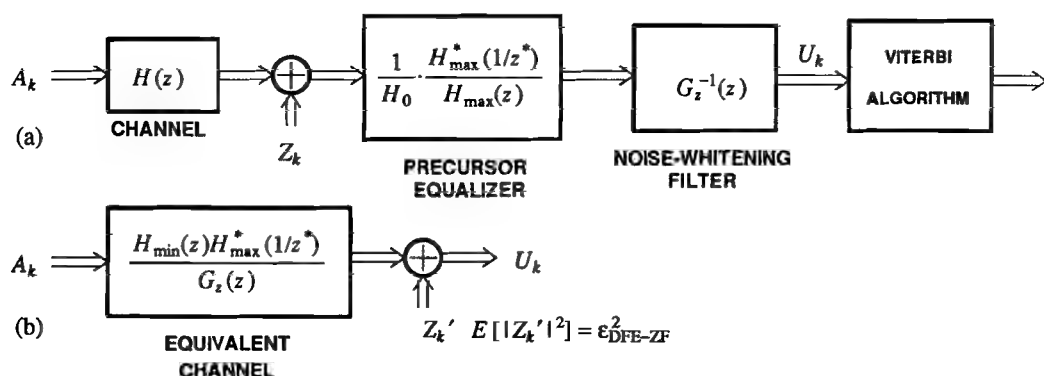
- Excluding zeros at  $z = 0$  and  $z = \infty$ ,  $G_z$  must have no zeros, implying that the noise spectrum  $S_z$  must be an all-pole spectrum (such a noise process is known as *autoregressive*).

It should be emphasized that the configuration of Figure 10-16a is not the ML sequence detector applied to the continuous-time received signal (in the sense derived in Chapter 8) unless of course the channel model  $H$  was obtained from the WMF receiver front end. However, if we do not use a WMF front end, the Viterbi algorithm applied as in Figure 10-16a is still a MLSD with respect to that discrete-time channel model, even though it is suboptimal with respect to the continuous-time model.

What do we do when  $E$  is not FIR? The VA remains useful in this case, even though it is not precisely an ML detector. There are two possible approximations we can use:

- The precursor equalizer can be modified to result in an FIR response, in which case the noise samples at the VA input will not be precisely white.
- The channel model can be truncated to an FIR response assumed in the realization of the VA, in which case there will be some residual ISI not considered in the VA.

Either of these approximations become more accurate as the number of taps in the artificially assumed FIR response becomes larger.



**Figure 10-16.** (a) The Viterbi algorithm as applied to the detection of data symbols in the presence of ISI. It must be assumed that the isolated pulse response at the precursor equalizer output is FIR. (b) Equivalent model, in which the channel is causal and the additive noise  $Z_k'$  is white with variance  $\epsilon_{DFE-ZF}^2$ .

### 10.3. FRACTIONALLY SPACED EQUALIZER

One of the surprising properties of the sampled MF and the WMF is the symbol-rate sampling. Assuming that the PAM signal design uses some excess bandwidth, as it must for practical reasons illustrated in Chapter 6, this sampling rate is less than would be required to avoid aliasing distortion. It seems strange that aliasing distortion would be deliberately introduced, and indeed there are some practical problems that result. Thus, it is common to use a higher sampling rate in a receiver front end and equalization, using an architecture known as the *fractionally spaced equalizer (FSE)* [10]. This fractionally spaced approach applies to all the receiver design strategies we have considered: the LE and DFE, WMF, and MLSD. A properly designed FSE is equivalent to the matched filter plus symbol-rate sampling, but offers considerable advantages [11,12,13]. First we will outline two of the problems addressed by the FSE, and then derive the FSE structure in two ways — in the time-domain and in the frequency-domain.

#### Aliasing and Sampling Phase

From a practical perspective, the assumptions under which the sampled matched filter was derived are problematic. There are several assumptions underlying this structure that are often violated in practice:

- The receiver sampling phase is precisely known relative to the symbol interval.
- The channel response  $h(t)$  is known precisely. This is partially overcome with the adaptive equalization techniques of Chapter 11, but the practical difficulty is the continuous-time matched filter, which is not easily adapted.
- The discrete-time filters, such as the linear equalizer or DFE precursor equalizer, have unconstrained complexity.

To see the effect of a sampling phase error, assume that the input isolated pulse is delayed in time by an unknown time  $t_0$  [14]. The complex-valued baseband pulse is then  $h(t - t_0)$  rather than  $h(t)$ , but the sampling times  $t = kT$  at the matched filter output are unchanged.

#### Exercise 10-3.

Verify that, with this time delay, the white-noise folded spectrum of an isolated pulse after sampling is

$$S_{h,t_0}(e^{j\omega T}) = \frac{1}{T} e^{j\omega t_0} \sum_{m=-\infty}^{\infty} |H(j(\omega + m\frac{2\pi}{T}))|^2 e^{j2\pi m \frac{t_0}{T}}. \quad (10.97)$$

□

While  $S_h(e^{j\omega T})$  is a non-negative real-valued function,  $S_{h,t_0}(e^{j\omega T})$  is no longer necessarily either real-valued or non-negative. In fact,  $S_{h,t_0}(e^{j\omega T})$  can have spectral nulls or near-nulls that are not present in  $S_h(e^{j\omega T})$ . An equalizer that compensates for these nulls can produce considerably more noise enhancement than when  $t_0 = 0$ .

**Example 10-21.**

When there is less than 100% excess bandwidth, we get the simpler relation

$$S_{h,t_0}(e^{j\omega T}) = e^{j\omega t_0} |H(j(\omega))|^2 (1 + e^{-j2\pi \frac{t_0}{T}} \alpha), \quad 0 \leq \omega \leq \frac{\pi}{T} \quad (10.98)$$

where

$$\alpha = \frac{|H(j(\omega - \frac{2\pi}{T}))|^2}{|H(j\omega)|^2}. \quad (10.99)$$

If the channel falls off monotonically, then  $\alpha$  will generally be less than unity. The term in parentheses in (10.98) has squared magnitude

$$|1 + e^{-j2\pi \frac{t_0}{T}} \alpha|^2 = 1 + \alpha^2 + 2\alpha \cos(2\pi \frac{t_0}{T}), \quad (10.100)$$

which has minimum value  $(1 - \alpha)^2$  when  $t_0 = T/2$ . The folded spectrum has a near-null at the frequency for which  $\alpha$  is maximum, and the depth of the null depends on  $t_0$ , with the worst case when  $t_0 = T/2$ . The folded spectrum depends on  $t_0$ , and for some values of  $t_0$  the noise enhancement can be much worse than for others.  $\square$

When equalizer filters are made adaptive (Chapter 11), their complexity must be constrained. The symbol-rate folded spectrum in a passband QAM channel can have rapid transitions near the band edges that are difficult to equalize using a constrained complexity equalizer [15].

A receiver front end that moves the matched filter to discrete time, which requires a higher than symbol-rate sampling, is equivalent to the sampled matched filter under idealized assumptions, but is superior under more realistic assumptions. This alternative receiver structure will be derived first in the time domain, and then in the frequency domain.

**Time Domain Derivation**

Assuming that the noise spectrum is white, the matched filter following demodulation equivalently correlates against the received pulse  $h^*(t)$ . For simplicity, assume that the channel has an excess bandwidth of less than 100%, so that  $h(t)$  is bandlimited to  $2\pi/T$  radians/sec. The sampling theorem (Section 2.3) tells us that the received pulse  $h(t)$  can be expanded in terms of its samples at twice the symbol rate,  $h_m = h(mT/2)$ ,

$$h(t) = \sum_{m=-\infty}^{\infty} h_m \operatorname{sinc}[\frac{2\pi}{T}(t - m\frac{T}{2})] \quad (10.101)$$

where  $\operatorname{sinc}(x) = \sin(x)/x$ . We can replace  $h(t)$  by (10.101) in calculating the folded spectrum, resulting in a sampled matched filter output of

$$\sum_{m=-\infty}^{\infty} h_m^* S_{2k+m} \quad (10.102)$$

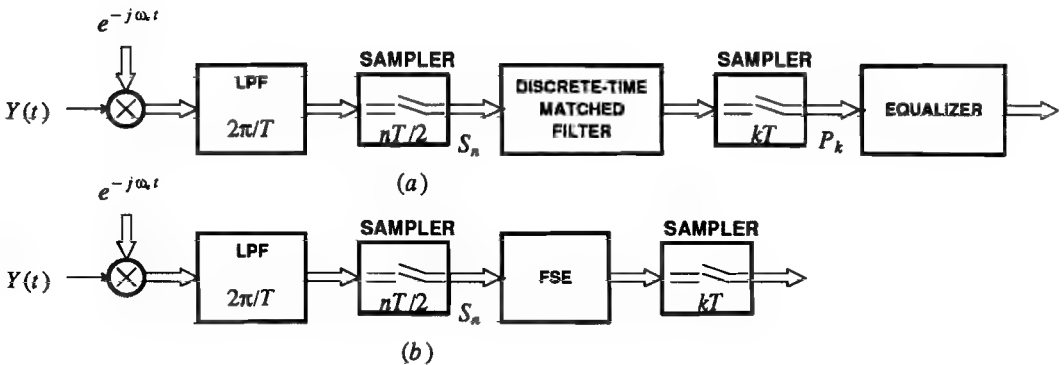
where

$$S_n = \int_{-\infty}^{\infty} Y(t) e^{-j\omega_c t} \text{sinc}\left[\frac{2\pi}{T}\left(t - n\frac{T}{2}\right)\right] dt \quad (10.103)$$

$S_n$ ,  $-\infty < n < \infty$  is evidently another sufficient statistic for the received signal since the matched filter output can be deduced from it using (10.102).

The sufficient statistics  $S_n$  in (10.103) are simply the sampled output of an ideal lowpass filter, where the filter bandwidth is  $2\pi/T$  as shown in Figure 10-17a. This lowpass filter rejects all out-of-band noise components, and especially the out-of-band noise that would otherwise alias inband after sampling. Since the output has bandwidth less than  $2\pi/T$ , by the sampling theorem it can be sampled at rate  $2/T$  Hz (twice the symbol rate) without loss of information. The calculation of the second sufficient statistic in (10.102) can be thought of as applying a matched filter to the lowpass filter output in discrete time. The output of this matched filter can be sampled at the symbol rate as before, so that this discrete-time matched filter also performs a decimation (by a factor of two) implicitly. This is followed by a symbol-rate equalizer, for example a linear equalizer or a precursor equalizer.

The two discrete-time filters in Figure 10-17a can be combined as in Figure 10-17b, where the resulting combined matched filter and equalizer is called a *fractionally spaced equalizer (FSE)*. It is a filter that has an input sampling rate equal to twice the symbol rate, and output sampling rate equal to the symbol rate. We can think of it as a filter with input and output sampling rates equal to twice the symbol rate, where every second output sample is not used and hence need not be calculated. The FSE structure shown in Figure 10-17b is applicable to any of the design strategies considered previously, including the sampled matched filter, WMF, LE, DFE, and MLSD. An FSE is assumed in Figure 6-23.



**Figure 10-17.** Fractionally spaced equalizer for 100% excess bandwidth. a. Realization with separate discrete-time matched filter and equalizer. b. Realization where the discrete-time matched filter and equalizer are combined.

**Exercise 10-4.**

Determine a formula for the output of the FSE before decimation in terms of the received pulse  $h(t)$  and equalizer  $C(z)$ , and thereby determine the transfer function of the FSE.  $\square$

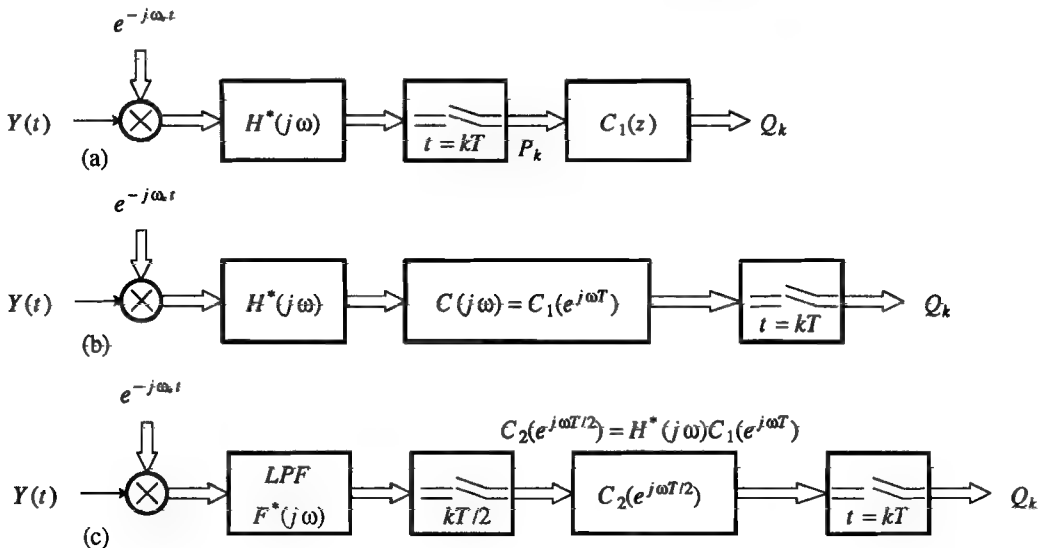
**Frequency Domain Derivation**

It is instructive to rederive the FSE entirely in the frequency domain. The conventional matched filter  $H^*(j\omega)$ , symbol-rate sampler, and a discrete-time equalizer  $C_1(z)$  are shown in Figure 10-18a. The first step in the derivation of the FSE is to implement the discrete-time filter equivalently in continuous time as a filter with transfer function  $C_1(e^{j\omega T})$ . This moves the symbol-rate sampler after the equalizer, but does not change the output.

**Exercise 10-5.**

Show mathematically that Figure 10-18a and Figure 10-18b are equivalent.  $\square$

Assume again that the excess bandwidth is less than 100% (this can be relaxed, see Problem 10-9). Then the combined filter  $H^*(j\omega)C_1(e^{j\omega T})$  has bandwidth less than  $2\pi/T$ , and hence can be implemented as a discrete-time filter with sampling rate equal to  $2/T$ , or twice the symbol rate. This is shown in Figure 10-18c; the channel output is first filtered by an anti-aliasing filter that eliminates all noise and signal



**Figure 10-18.** FSE interpretation in the frequency domain. a. The starting point is the conventional matched filter, symbol-rate sampler, and equalizer  $C_1(z)$ . b. The equalizer implemented in continuous time. c. The matched filter and equalizer implemented in discrete time with sampling rate equal to twice the symbol rate.



components above the symbol rate  $1/T$ , a twice symbol-rate sampler, and a discrete-time filter that realizes both the matched filter and the precursor equalizer. The output of this discrete-time filter is resampled at the symbol rate, implying that this is a decimating filter (Problem 10-10).

### Interpretation

The symbol-rate sampling after a matched filter usually introduces aliasing. In both the time and frequency domains, we have seen that the FSE uses a discrete-time filter at a sampling rate such that there is no aliasing before filtering. The result is that the discrete-time filter can be designed to adjust for the sampling phase in the matched filtering portion of its response, thereby eliminating the effect of sampling phase on noise enhancement. In practice, we will find in Chapter 11 that the discrete-time FSE can adapt automatically to compensate for the sampling phase, thereby reducing the effect of sampling phase on noise enhancement. In addition, the limited degrees of freedom of a finite discrete-time filter are more effective when deployed before decimation to the symbol rate, because the aliased sidebands can be filtered independently.

It should be remembered that a twice-symbol-rate FIR FSE with the same number of coefficients as a symbol-rate discrete-time filter has an impulse response that will span half the time interval. In spite of this, experience has shown that the FSE will perform as well for the same number of coefficients for all channel conditions, and noticeably better for channels with severe band-edge delay distortion [12].

We do not need to redo the equalization designs of Sections 10.1 and 10.2, because of the equivalence to the FSE shown in Figure 10-18. Thus, a simple approach is to design the equalizers according to the theory of Sections 10.1 and 10.2, and then transfer the resulting design to the fractionally spaced implementation using the equivalence property. Note that fractionally spaced equalizers can be used for the LE, the WMF, and the precursor equalizer of the DFE. However, the postcursor equalizer of the DFE will always use symbol-rate sampling, because it operates at the same sampling rate as the slicer.

## 10.4. TRANSVERSAL FILTER EQUALIZERS

In previous sections of this chapter, we have considered equalizer designs without regard to implementation complexity. In practice any equalizer that is built must be realizable, which means that it must have a rational transfer function. Rational transfer functions come in two basic types, finite impulse response (FIR) or infinite impulse response (IIR). FIR filters can be implemented even if they are not causal, if an additional delay is permissible. Such a delay will be permissible in the feedforward, but not feedback, path. IIR filters can be implemented as stable and causal filters as long as their poles are inside the unit circle.

If the folded spectrum  $S_h(e^{j\omega T})$  happens to be rational, then all the equalizer designs are rational. However, it is very unusual for a channel to have *precisely* a rational transfer function, so we must resort to an approximation. Moreover, we saw

in Section 10.2 that equalizer designs resulting from optimization often have poles outside the unit circle, when the discrete-time channel model has a maximum-phase component. These equalizers can only be approximated, usually by an FIR filter. The usual approximation would be something like

$$C(z) = \sum_{k=-N}^N c_k z^{-k}, \quad (10.104)$$

which is non-causal. In fact an equalizer of the form of (10.104) is realizable if the output is delayed by  $N$  symbol intervals, since

$$z^{-N} C(z) = \sum_{k=0}^{2N} c_{k-N} z^{-k} \quad (10.105)$$

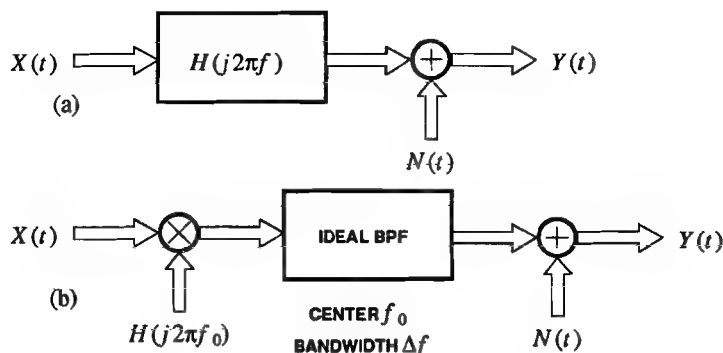
is causal. Of course any physically realizable system must be strictly causal, but we have assumed a non-causal matched filter, which again can be approximated by a causal filter plus delay.

In digital communication, an FIR digital filter is usually given the special name *transversal filter*, a terminology that originated in early continuous-time realizations using analog delay lines. The coefficients  $c_k$  of the filter are usually called the *tap weights* or *tap coefficients* of the filter. In the case of (10.104) there are precisely  $2N + 1$  taps for the filter, which gives us  $2N + 1$  degrees of freedom in the design of the filter.

Optimization of equalizer designs based on a constrained-complexity filter such as (10.104) is feasible, although quite different techniques than those used earlier in this chapter are necessary. The equalizer transfer function can no longer be chosen independently at each frequency as was done earlier. This implies that the zero-forcing criterion is no longer feasible, since a constrained-complexity filter may not be able to force the ISI to zero. Thus, a minimum MSE criterion is often used, and the MSE is minimized over the choice of filter tap coefficients (considered as a  $(2N+1)$ -dimensional vector). An example of this type of optimization will arise in Chapter 11, as a first step in the design of adaptive equalizers.

## 10.5. ISI and CHANNEL CAPACITY

In this chapter we have established the effect of ISI on the error probability of several receiver designs, including both equalization and the MLSD. Another related question is the effect of ISI on channel capacity. While the error probability results allow us to predict the impact of ISI when channel coding is not used, the channel capacity results allow us to assess the effect of ISI in the presence of channel coding. In Section 8.7 we did a similar development for ideal channels without ISI, and found the performance of different modulation techniques in relation to the fundamental limits of capacity in the absence of ISI. It was found that a given modulation technique at a given probability of error displayed an "SNR gap to capacity", which is the difference between the normalized SNR required to achieve that error probability and the



**Figure 10-19.** Continuous-time channel models for the calculation of capacity. (a) A continuous-time channel with additive Gaussian noise. (b) A small subband of the channel.

normalized SNR that the channel capacity theorem says is required for a vanishingly small error probability. Our purpose here is to extend those results to channels with ISI.

Since implementation of processing techniques is easier in discrete time, we must choose a sampling rate and transmit and receive filters that effectively transmit a discrete-time signal over a continuous-time channel. If the continuous-time channel is not bandlimited, then this conversion to discrete time sometimes involves a reduction in capacity, although, as we will see, that is more likely to occur at high SNR. For the most part, we will apply our discrete-time techniques to continuous-time channels that are strictly bandlimited, because this case is analytically much simpler. (At the end of the section, some differences in the conclusions for channels that are not strictly bandlimited will be mentioned.) For strictly bandlimited channels we will find some surprising conclusions, including:

- ISI always reduces the capacity of a channel. This is surprising because we have seen that ISI does not always appreciably increase the error probability for PAM, for example on some channels with mild ISI when the MLSD achieves a figure of merit equal to the matched filter bound.
- On channels with ISI, we will find a generalized definition of normalized SNR,  $SNR_{\text{norm}}$ , that has the same interpretation as  $SNR_{\text{norm}}$  in Section 8.7; namely,  $SNR_{\text{norm}} \geq 1$ , with equality when the modulation technique is operating at capacity limits. The difference between  $SNR_{\text{norm}}$  and unity represents an "SNR gap to capacity", or increase in transmitted power or SNR relative to the minimum transmitted power or SNR at which it is feasible to operate, as quantified by the channel capacity.
- When we use PAM in conjunction with the equalization strategies considered in Section 10.1, we will quantify the SNR gap. For the DFE-ZF, we will find that this SNR gap approaches a constant at high SNR that is *independent of the channel response*. Since an ISI-free channel is a special case of a channel with ISI, this asymptotic SNR gap is, at high SNR, the same on channels with ISI as on

ideal channels. This implies that channel coding (Chapter 14) has approximately the same potential for improvement in the normalized SNR at a given error probability on channels with ISI as on ideal channels. This does *not* imply that ISI is benign with respect to channel capacity, since it does reduce channel capacity as noted above, but rather it *does* imply that the gap between coded modulation systems and capacity can be closed essentially to the same extent on channels with ISI as on ideal channels, at least at high SNR. Furthermore, it suggests that the DFE-ZF, and not the MLSD, is a good receiver structure to use as a starting point.

- The SNR gap for the DFE-MSE-U is fixed, independent of the ISI, at all SNR's (not only at high SNR as with the DFE-ZF), as long as the transmit filter is optimized. This suggests that the unbiased DFE-MSE may be a good canonical receiver structure for coded modulation systems at low SNR.

### 10.5.1. Continuous-Time Water Pouring

As in Section 8.7, the first step will be to determine the capacity of the continuous-time additive Gaussian noise channel. We do this first for a general channel, and then apply the results to the special case of a passband channel.

Suppose, as pictured in Figure 10-19, that we have a real-valued channel with transfer function  $H(j2\pi f)$  and additive real-valued Gaussian noise  $N(t)$ . (In dealing with capacity, we will find it generally advantageous to use frequency variable  $f$  rather than  $\omega$ , as is prevalent elsewhere in this book, because many factors of  $2\pi$  go away.) While we are primarily interested in the white noise case, the following development is no harder if we assume that the noise has a general power spectrum  $S_N(j2\pi f)$ . In Section 8.7 we developed the channel capacity for a similar situation, except the channel was an ideal channel with bandwidth  $B$  Hz. We were then able to derive the fundamental limits to the spectral efficiency of this ideal channel. Our purpose here is to extend these results to channels with ISI, and hence general  $H(j2\pi f)$  and  $S_N(j2\pi f)$ .

The earlier ideal channel results can be applied to the present situation by dividing bandwidth into small bins of width  $\Delta f$ , where  $\Delta f$  is small enough that the channel transfer function is approximately constant over a range of  $\Delta f$ . (Of course, later we will take  $\Delta f \rightarrow 0$ , at which time the approximation will become precise.) One of these subbands, centered at frequency  $f_0$ , is shown in Figure 10-19b. This subband is modeled by a flat gain of  $H(j2\pi f_0)$  together with an ideal (unit gain) real-valued bandpass filter with bandwidth  $\Delta f$  centered at frequency  $f_0$ . Even though the additive noise is not white, it can be considered as white with power spectral density  $S_N(j2\pi f_0)$  within the bandwidth of interest in the subchannel. Since each of these subbands can be used in principle by a separate and independent digital communication system, the total capacity is the aggregate capacity of each of these subbands.

Suppose there is a constraint on the average transmit power  $E[X^2(t)] = P_S$ . Furthermore, let  $S_X(j2\pi f)$  be an appropriate input power spectrum that meets this constraint. Choosing  $S_X(j2\pi f)$  is equivalent to deciding how to distribute the available input power across frequencies, or across subbands. Our approach is to determine the capacity of each subband, assuming its input power is constrained by

$S_X(j2\pi f)$  (at the frequency of that subband), giving us an expression for the total capacity as a function of  $S_X(j2\pi f)$ . Subsequently, the  $S_X(j2\pi f)$  that maximizes this total capacity will be found, subject to the constraint that  $E[X^2(t)] = P_S$ .

The capacity of one subband can be determined as follows. The constraint on the power into the subband is  $2S_X(j2\pi f_0)\Delta f$ , taking into account both positive and negative frequencies. This power and the gain factor  $H(j2\pi f_0)$  is equivalent to an ideal (unity transfer function) subchannel with input power constrained to  $2S_X(j2\pi f_0)|H(j2\pi f_0)|^2\Delta f$ . The capacity of this subchannel is given by (8.137),

$$\begin{aligned} C(j2\pi f_0) &= \Delta f \cdot \log_2 \left[ 1 + \frac{2S_X(j2\pi f_0)|H(j2\pi f_0)|^2\Delta f}{2S_N(j2\pi f_0)\Delta f} \right] \\ &= \Delta f \cdot \log_2 \left[ 1 + \frac{S_X(j2\pi f)|H(j2\pi f)|^2}{S_N(j2\pi f)} \right]. \end{aligned} \quad (10.106)$$

The total capacity is then the sum of the subchannel capacities, which becomes an integral in the limit of  $\Delta f \rightarrow 0$ ,

$$\begin{aligned} C &= \int_0^\infty \log_2 \left[ 1 + \frac{S_X|H|^2}{S_N} \right] df \\ &= \frac{1}{2} \int_{-\infty}^\infty \log_2 \left[ 1 + \frac{S_X|H|^2}{S_N} \right] df \quad \text{bits/sec.} \end{aligned} \quad (10.107)$$

In this and many subsequent expressions the frequency variable is suppressed for compactness.

Since capacity in each subband is achieved by a wide-sense stationary Gaussian input process, total capacity is achieved if  $X(t)$  is wide-sense stationary Gaussian with power spectrum  $S_X(j2\pi f)$ . What remains is to translate an overall input power constraint into an optimal power spectrum  $S_X(j2\pi f)$ . Our desire is to maximize  $C$  under the constraints

$$P_S = \int_{-\infty}^\infty S_X df, \quad S_X \geq 0, \quad (10.108)$$

where  $P_S$  is the transmitted power. Using a Lagrange multiplier approach to meet the power constraint, we can equivalently choose  $S_X$  as a function of  $\lambda$  to maximize

$$\frac{1}{2} \int_{-\infty}^\infty \log_2 \left[ 1 + \frac{S_X|H|^2}{S_N} \right] df + \lambda \int_{-\infty}^\infty S_X df \quad (10.109)$$

and then choose  $\lambda$  to meet power constraint (10.108). Writing this as a single integral and differentiating the integrand with respect to  $S_X$  at each frequency and setting that derivative to zero, we get that  $S_X$  is of the form  $(1/2\lambda - S_N/|H|^2)$ . Taking into account that  $S_X$  must be positive, the result is

$$S_X = \begin{cases} L - \frac{S_N}{|H|^2}, & f \in F \\ 0, & \text{otherwise,} \end{cases} \quad (10.110)$$

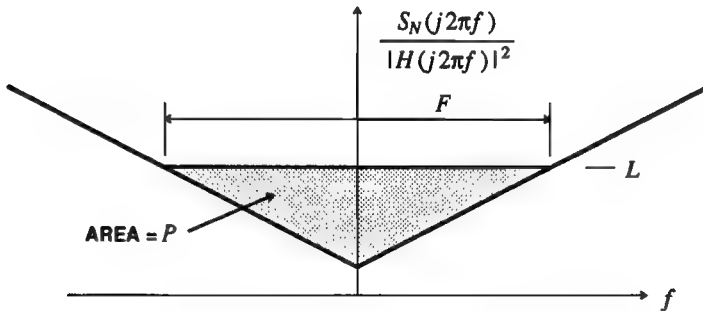
where the constant  $L = 1/2\lambda$  is chosen to meet the power constraint and  $F$  is the set of frequencies for which  $L > S_N/|H|^2$ .  $F$  is called the *water-pouring band*. This method of determining the transmit spectrum is called *water pouring*, and we will call the resulting transmit spectrum  $S_X$  the *water-pouring spectrum* [16].

The approach of splitting up the overall channel into subchannels also points out that the input random process that achieves capacity is a wide-sense stationary (and hence strictly stationary) real-valued Gaussian process with power spectral density  $S_X(j2\pi f)$ . Thus, communication systems that wish to approach capacity have to somehow ensure that the transmit spectrum approximates the water-pouring spectrum, and also that the distribution is approximately Gaussian.

This solution is easiest to understand pictorially, as in Figure 10-20. A "bowl"  $S_N/|H|^2$  is formed, and water is poured into the bowl up to level  $L$  until the total volume of water equals the transmit power constraint. The transmit spectrum vs. frequency is then the depth of the water. Water pouring concentrates the transmit power at those frequencies where the ratio  $|H|^2/S_N$  is relatively large; that is, the channel has relatively little attenuation or the noise power spectrum is relatively small.

### Passband Channel

Observe that the water-pouring approach applies equally well to passband as well as baseband channels.



**Figure 10-20.** An illustration of water pouring for the optimization of the transmit power spectrum in achieving capacity. A "bowl" shaped like  $S_N(j2\pi f)/|H(j2\pi f)|^2$  is formed, and water with volume  $P_S$  is poured into it up to level  $L$ . The capacity-achieving transmit spectrum at each frequency is the depth of the water.

**Example 10-22.**

For an ideal passband white Gaussian noise channel with bandwidth  $B$ ,  $S_N/|H|^2 = N_0$  is a constant within the channel bandwidth. The bowl (actually two bowls) have infinite-slope sides, as illustrated in Figure 10-21, because  $|H|^2 = 0$  outside the channel bandwidth. Thus, the water-pouring band is precisely the full passband bandwidth. The volume in one of the bowls must be  $P_S/2$ , which implies that the depth of the water must be  $P_S/2B$ ; that is,  $S_X = P_S/2B$  over the bandwidth of the channel. The capacity is then

$$C = B \cdot \log_2(1 + \text{SNR}), \quad \text{SNR} = P_S/2BN_0, \quad (10.111)$$

consistent with (8.137).  $\square$

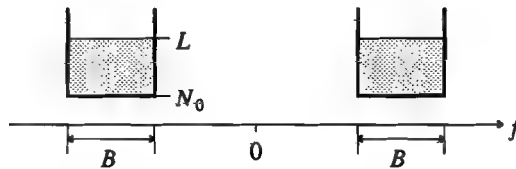
A formula for capacity in terms of a complex baseband equivalent channel can be derived for a passband channel. Assume the channel  $H$  is strictly bandlimited to  $f_0 - B/2 \leq |f| \leq f_0 + B/2$ . Then the capacity integral of (10.107) can be safely limited to this frequency range, since the water-pouring band must fall in this range. Thus, (10.107) can be written as the sum of two integrals, one for positive and one for negative frequencies, or equivalently twice the positive-frequency integral due to the symmetry of  $|H|$  about  $f = 0$  (since the channel  $H$  is real-valued). Thus,

$$\begin{aligned} C &= \int_{f_0 - B/2}^{f_0 + B/2} \log_2 \left[ 1 + \frac{S_X(j2\pi f) |H(j2\pi f)|^2}{S_N(j2\pi f)} \right] df \\ &= \int_{-B/2}^{B/2} \log_2 \left[ 1 + \frac{S_X(j2\pi(f + f_0)) |H(j2\pi(f + f_0))|^2}{S_N(j2\pi(f + f_0))} \right] df. \end{aligned} \quad (10.112)$$

When water pouring is performed at passband, it is clear from symmetry that half the power will be at positive frequencies, and thus we can formulate water pouring at baseband using  $P_S/2$  in place of  $P_S$ ,

$$S_X(j2\pi(f + f_0)) = \begin{cases} L - \frac{S_N(j2\pi(f + f_0))}{|H(j2\pi(f + f_0))|^2}, & f \in F \\ 0, & \text{otherwise,} \end{cases} \quad (10.113)$$

where  $L$  is chosen such that



**Figure 10-21.** Illustration of the water pouring procedure for the ideal bandpass white Gaussian noise channel.

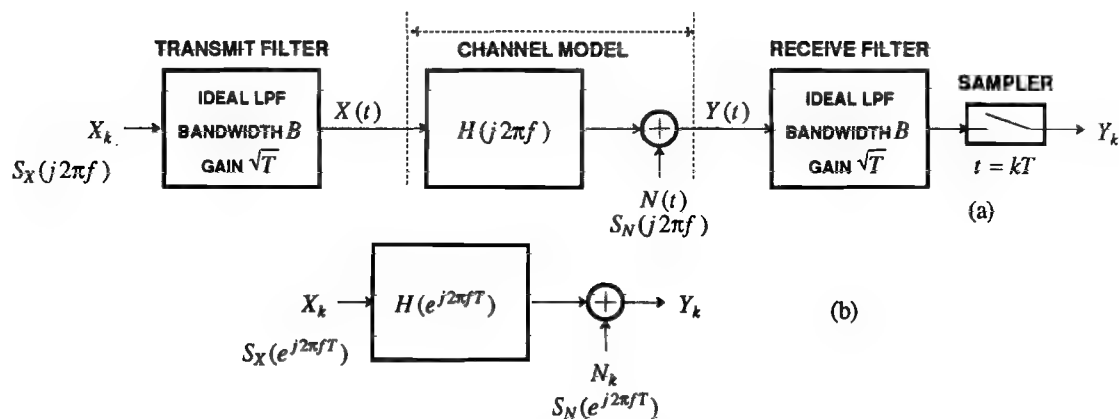
$$\int_{-B/2}^{B/2} S_X(j2\pi(f + f_0)) df = \frac{P_S}{2}. \quad (10.114)$$

Also,  $F$  must be a subset of  $[-B/2, B/2]$ . The complex baseband channel does not necessarily have spectra symmetric about  $f = 0$  (the baseband-equivalent channel and noise are complex-valued), and thus the water-pouring band  $F$  is not necessarily symmetric about  $f = 0$ .

### 10.5.2. Discrete-Time Water Pouring

When a continuous-time channel is strictly bandlimited, and we derive a discrete-time channel from it using a sufficiently high sampling rate, then the discrete-time channel will have the same capacity as the underlying continuous-time channel. This will also be true of channels which are not bandlimited, but for which the transfer function approaches zero as frequency gets large. We will now form the connection between the channel capacities of discrete- and continuous-time channels, leading to a formula for the capacity of a discrete-time channel. We will gain the ability to compare the capacity of this channel with the performance of PAM in conjunction with receiver design techniques covered earlier in this chapter.

Assume that  $H(j2\pi f)$  is a real-valued continuous-time channel, which may or may not be bandlimited and may be baseband or passband. A discrete-time channel is derived in Figure 10-22a by sampling at rate  $2B = 1/T$  at both input and output with ideal LPF transmit and receive filters each with gain  $\sqrt{T}$ , to ensure that their impulse response has unit energy. The resulting discrete-time system is pictured in Figure 10-22b. We will now relate the discrete-time transfer function and power spectra to the similar quantities for the continuous-time channel.



**Figure 10-22.** The way in which a discrete-time system can be obtained from a continuous-time system employing the sampling theorem. (a) Deriving the discrete-time channel by sampling at the input and output. (b) The resulting discrete-time channel model.



### Signal Transfer Function

Since the receive filter is bandlimited to half the sampling rate, there is no aliasing distortion in the final sampling operation. The overall gain of  $T$  in the transmit and receive filters makes up for the factor of  $1/T$  in the conversion from continuous to discrete time, and hence the discrete-time and continuous-time transfer functions are directly related,

$$H(e^{j2\pi fT}) = H(j2\pi f), \quad |f| \leq B. \quad (10.115)$$

### Transmit Power Constraint

The discrete- and continuous-time power spectra are related by

$$S_X(j2\pi f) = S_X(e^{j2\pi fT}), \quad |f| \leq B, \quad (10.116)$$

since there is a factor of  $1/T$  in the conversion from discrete- to continuous-time (see Appendix 3-A) compensated by the  $\sqrt{T}$  gain of the filter. The relation

$$E[X_k^2] = T \cdot \int_{-1/2T}^{1/2T} S_X(e^{j2\pi fT}) df = T \int_{-1/2T}^{1/2T} S_X(j2\pi f) df = T \cdot E[X^2(t)] \quad (10.117)$$

implies that a continuous-time power constraint  $P_S = E[X^2(t)]$  is equivalent to a discrete-time constraint on the energy of a single sample,  $P_S \cdot T = E[X_k^2]$ . Furthermore, any continuous-time power spectrum  $S_X(j2\pi f)$  can be generated, *providing that it is bandlimited to  $B$  Hz*. Thus, if capacity for the continuous-time channel demands a transmit power spectrum wider than this, in accordance with water pouring, then it cannot be generated by the configuration of Figure 10-22a for sampling rate  $2B$ . Conversely, if the bandwidth of the continuous-time water-pouring band is less than  $B$ , then the discrete-time channel capacity will equal the continuous-time channel capacity, with the transmit power spectrum chosen appropriately.

### Output Noise Spectrum

When the power spectrum of the noise on the continuous-time channel is  $S_N(j2\pi f)$ , then the noise at the output of the receive filter is bandlimited to  $B$  Hz and multiplied by  $T$ . The sampling operation does not introduce aliasing distortion, and multiplies the power spectrum by  $1/T$ , and thus after sampling

$$S_N(e^{j2\pi fT}) = S_N(j2\pi f), \quad |f| \leq B. \quad (10.118)$$

### Capacity of the Discrete-Time System

The capacity of the discrete-time system with input power constraint  $E[X_k^2] = P_S$  will equal the capacity of the continuous-time system with input power constraint  $E[X^2(t)] = P_S/T$ , providing that the resulting water-pouring band of the continuous-time system is contained in  $|f| \leq B$ . This will always be true if the continuous-time system is strictly bandlimited to  $B$ . Therefore, by assuming that  $H(j2\pi f)$  is bandlimited to  $B$  and satisfies (10.115) for  $|f| \leq B$ , we can calculate the capacity of the discrete-time system indirectly by calculating the capacity of the continuous-time system with input power constraint  $P_S$ . Thus, the capacity of a real-valued discrete-time

baseband channel  $H(e^{j2\pi fT})$  with additive real-valued noise  $S_N(e^{j2\pi fT})$  is

$$C = \frac{1}{2} \int_{-1/2T}^{1/2T} \log_2 \left[ 1 + \frac{S_X(e^{j2\pi fT}) |H(e^{j2\pi fT})|^2}{S_N(e^{j2\pi fT})} \right] df, \quad \text{bits/sec}, \quad (10.119)$$

where  $S_X(e^{j2\pi fT})$  is given by the water-pouring formula

$$S_X(e^{j2\pi fT}) = \begin{cases} L - \frac{S_N(e^{j2\pi fT})}{|H(e^{j2\pi fT})|^2}, & f \in F \\ 0, & \text{otherwise} \end{cases}, \quad (10.120)$$

where  $L$  is chosen such that the transmit power  $P_S = E[X_k^2]/T$  is

$$P_S = \int_{-1/2T}^{1/2T} S_X(e^{j2\pi fT}) df. \quad (10.121)$$

### Passband Case

We can use the capacity of a continuous-time passband channel, given by (10.112), to derive the capacity of a discrete-time complex-valued baseband channel derived from that passband channel. The connection is the modulation and demodulation step pictured in Figure 10-23. In Figure 10-23a, the factors of  $\sqrt{2}$  in the transmitter and receiver ensure that the powers of the two baseband and the single passband signal are the same. The resulting complex baseband channel is shown in Figure 10-23b. As shown in Chapter 6, the equivalent baseband channel is  $H(j2\pi(f + f_0))$ , and the equivalent baseband power spectrum is  $S_Z(j2\pi f) = 2 \cdot S_N(j2\pi(f + f_0))$ . The factor of two in  $S_Z$  arises mathematically from the  $\sqrt{2}$  in the demodulator, but can also be interpreted intuitively in two ways:

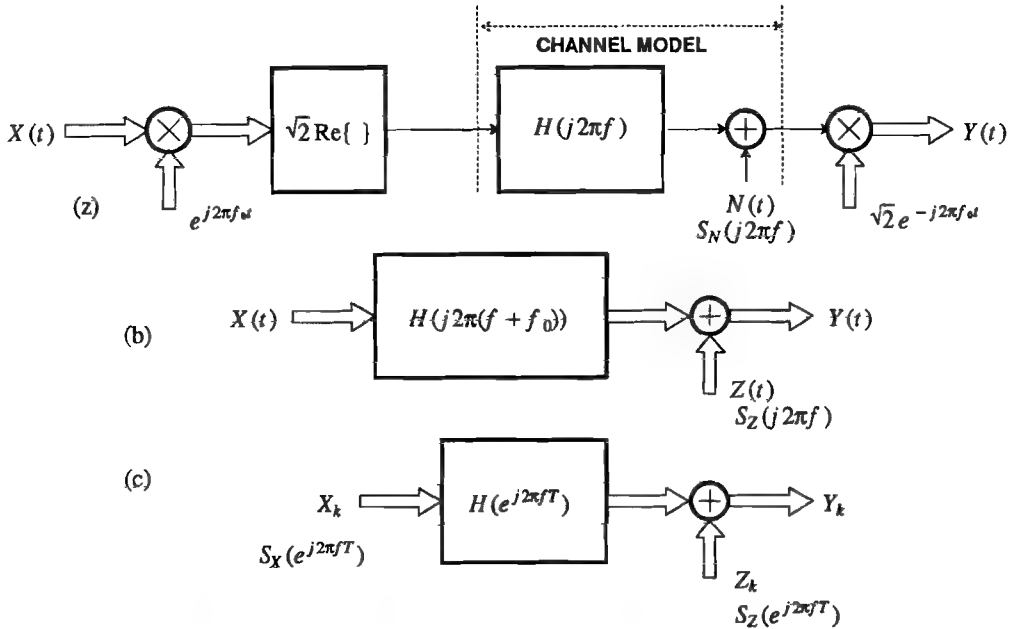
- The noise power spectrum doubles due to spreading the same noise over half the bandwidth at baseband compared to passband.
- The noise is divided between the real and imaginary parts in equal variance (due to circular symmetry of the complex Gaussian noise), each part with the same power spectrum (referred to baseband) as in the passband case, but half the bandwidth.

Finally, due to the  $\sqrt{2}$  factor in the transmitter real part, the relation between discrete-time and continuous-time signal power spectra is  $S_X(e^{j2\pi fT}) = 2 \cdot S_X(j2\pi(f + f_0))$ , which normalizes the total power to be the same at passband and baseband, and also results in the power constraint, from (10.114),

$$P_S = \int_{-1/2T}^{1/2T} S_X(e^{j2\pi fT}) df, \quad (10.122)$$

corresponding to power constraint  $P_S = E[|X_k|^2]/T$ .

Substituting these values into (10.112), the capacity of the complex baseband discrete-time channel is



**Figure 10-23.** The derivation of a complex baseband model from a passband channel  $B(j2\pi f)$ . (a) A continuous-time bandpass channel. (b) An equivalent complex baseband channel derived using modulator and demodulator. (c) A discrete-time complex baseband channel derived by applying the sampling and transmit/receive filter of Figure 10-22 to the baseband channel of (b).

$$C = \int_{-1/2T}^{1/2T} \log_2 \left[ 1 + \frac{S_X(e^{j2\pi fT}) |H(e^{j2\pi fT})|^2}{S_Z(e^{j2\pi fT})} \right] df, \quad \text{bits/sec}, \quad (10.123)$$

where the functional form of the transmit spectrum is, from (10.113),

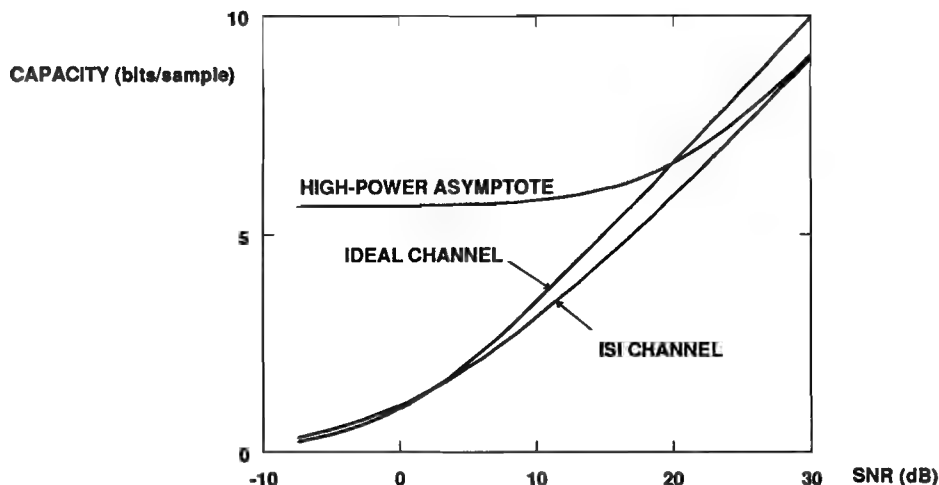
$$S_X(e^{j2\pi fT}) = \begin{cases} 2L - \frac{S_Z(e^{j2\pi fT})}{|H(e^{j2\pi fT})|^2}, & f \in F \\ 0, & \text{otherwise,} \end{cases} \quad (10.124)$$

and  $2L$  is chosen to meet constraint (10.122).

#### Example 10-23.

When the noise on the continuous-time channel is white with power spectrum  $N_0$ ,  $S_Z = 2N_0$ . It is convenient to define a normalized transmit spectrum,

$$S_X' = \frac{S_X}{2N_0} = \begin{cases} L' - |H|^{-2}, & f \in F \\ 0, & \text{otherwise,} \end{cases} \quad (10.125)$$



**Figure 10-24.** The capacity per sample is plotted against SNR ( $10 \log_{10} P_S / 2N_0B$ ) for a single-zero channel with  $|c| = 0.99$ . The capacity is independent of the angle of  $c$ .

in which case  $L'$  is chosen to meet the power constraint of (10.122),

$$\frac{P_S}{2N_0B} = T \cdot \int_{-1/2T}^{1/2T} S_X' df \quad (10.126)$$

and the capacity of (10.123) becomes

$$C \cdot T = T \cdot \int_{-1/2T}^{1/2T} \log_2 \left[ 1 + S_X' |H|^2 \right] df. \quad (10.127)$$

Of course,  $P_S / 2N_0B$  is familiar as the SNR of the ideal white Gaussian noise channel. These relations are convenient for calculating the capacity for this white noise case. If the variable of integration is changed from  $f$  to  $\theta = fT$ , the factors of  $T$  multiplying the integral conveniently go away. For example,

$$C \cdot T = \int_{-1/2}^{1/2} \log_2 \left[ 1 + S_X'(e^{j2\pi\theta}) \cdot |H(e^{j2\pi\theta})|^2 \right] d\theta. \quad (10.128)$$

Since  $1/T$  is the sampling rate in Hz, the quantity  $C \cdot T = C/B$  is the capacity per sample.  $\square$

#### Example 10-24.

Consider a single-zero channel  $H(z) = (1 - cz^{-1})/\sqrt{1 + |c|^2}$ , normalized to have a unit-energy impulse response. The capacity is plotted in Figure 10-24 for three cases: the actual channel with  $|c| = 0.99$ , the ideal channel ( $c = 0$ ), and a formula derived later that applies asymptotically at high SNR. For our present purposes, the interesting comparison is between the ideal channel and the channel with ISI. At low SNR, the channel with ISI actually has a higher capacity than the ideal channel, because the water-pouring spectrum can be concentrated at frequencies where the channel transfer function has gain. At high SNR, there is a modest penalty in capacity due to ISI.  $\square$

**Example 10-25.**

The capacity of the single-pole channel  $H(z) = \sqrt{1 + |c|^2} / (1 - cz^{-1})$  for  $|c| = 0.99$  is plotted in Figure 10-25. This channel has much more severe ISI than the single-zero channel, and as a result there is a much greater penalty in capacity due to ISI.  $\square$

**10.5.3. Relation to Arithmetic and Geometric Means**

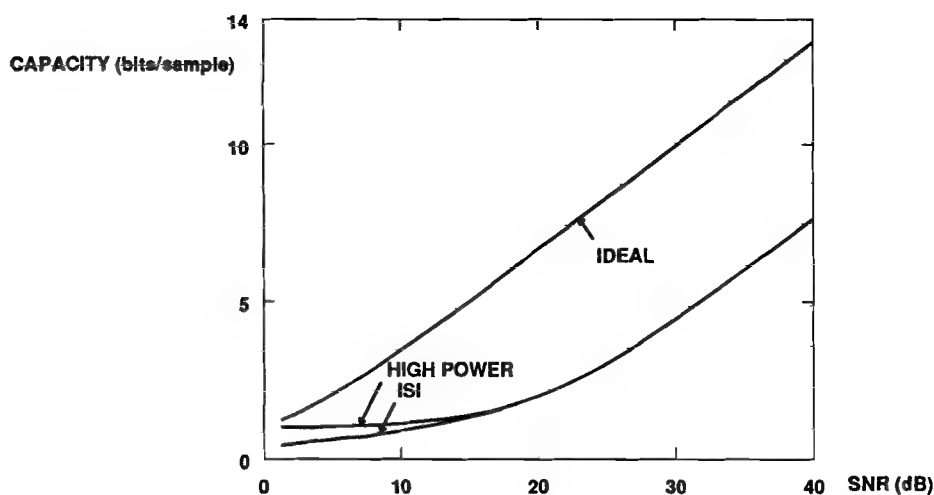
Once the water-pouring band  $F$  is determined, the capacity formulas can all be simplified in a useful way in terms of the arithmetic and geometric means. This gives a compact formula for the capacity, with the caveat that the formula is not self-contained, in that water-pouring must be performed first to determine  $F$ . Although this applies to both the continuous-time and discrete-time cases, we are primarily interested in discrete-time here. We will do the baseband case first, followed by the complex baseband equivalent to a passband channel.

**Baseband**

Assume we know  $F$ . From (10.120), we can calculate the total power for the water-pouring spectrum directly by integrating  $S_X$ , and restricting the integral to  $F$ ,

$$\frac{P_S}{|F|} = \langle S_X \rangle_{A,F} = L - \langle S_N / |H|^2 \rangle_{A,F}, \quad (10.129)$$

which gives us an equation for  $L$  in terms of the power constraint  $P_S$ . Also, substituting (10.120) into (10.119), where the integral is restricted to  $f \in F$ ,



**Figure 10-25.** The capacity per sample plotted against SNR ( $10 \log_{10} P_S / 2N_0 B$ ) for a single-pole channel with  $|c| = 0.99$ . The capacity is independent of the angle of  $c$ .

$$C = \frac{1}{2} \int_F \log_2 \left[ \frac{L \cdot |H|^2}{S_N} \right] df. \quad (10.130)$$

This is directly related to the geometric mean,

$$C = \frac{|F|}{2} \log_2 \langle L |H|^2 / S_N \rangle_{G,F} = \frac{|F|}{2} \log_2 \left[ \frac{L}{\langle S_N / |H|^2 \rangle_{G,F}} \right]. \quad (10.131)$$

Finally, substituting from (10.129),

$$C = \frac{|F|}{2} \log_2 \left[ \frac{P_S / |F| + \langle S_N / |H|^2 \rangle_{A,F}}{\langle S_N / |H|^2 \rangle_{G,F}} \right] \text{ bits/sec.} \quad (10.132)$$

#### Example 10-26.

When  $S_N = N_0$  is white noise, the capacity becomes

$$C = \frac{|F|}{2} \cdot \log_2 \left[ \frac{SNR + \langle |H|^{-2} \rangle_{A,F}}{\langle |H|^{-2} \rangle_{G,F}} \right] \text{ bits/sec,} \quad (10.133)$$

where  $SNR = P_S / |F| N_0$  is the channel *input* signal-to-noise ratio within the water-pouring band  $F$  ( $|F|$  is the total bandwidth of the water-pouring band in Hz, and when multiplied by  $N_0$  we get the total noise within this bandwidth). Note that the signal-to-noise ratio at the channel *output* is dependent on the channel transfer function. Equation (10.133) bears a striking relationship to (10.111), and of course when there is no ISI ( $H = 1$  for  $f \in F$ ) it reduces to (10.111).  $\square$

Once the water-pouring band  $F$  is determined, the only parameters of the channel that need be known are the arithmetic and geometric means of  $S_N / |H|^2$  over  $F$ . All channels with the same water-pouring band, and the same arithmetic and geometric means over that bandwidth, have the same capacity. This result is striking in view of the fact that the arithmetic and geometric means (over the entire bandwidth) determine the performance of the equalizers considered in Section 10.2 as well.

#### Complex Baseband Case

The only differences in the complex baseband case are the replacement of  $L$  by  $2L$  in the water-pouring formula (which doesn't change the result since it is just another constant), the replacement of  $S_N$  by the complex noise  $S_Z$ , and the removal of the factor of  $1/2$ . Thus, the capacity for the complex baseband case is similar,

$$C = |F| \cdot \log_2 \left[ \frac{P_S / |F| + \langle S_Z / |H|^2 \rangle_{A,F}}{\langle S_Z / |H|^2 \rangle_{G,F}} \right] \text{ bits/sec.} \quad (10.134)$$

The biggest difference is a capacity that is a factor of two larger, due to the complex signals; that is, the capacity *per dimension* stays the same. Of course, other differences in practice include a 50% smaller bandwidth (compensating for the higher dimensionality), and spectra that are generally not symmetric about  $f = 0$ .

### 10.5.4. Spectral Efficiency, SNR, and Normalized SNR

Thus far in this section, the capacity of both continuous- and discrete-time channels with additive Gaussian noise have been determined. These results will now be applied to characterizing the performance of digital communication systems operating over such channels. The results of Section 8.7, in which the operation of specific modulation techniques were related to the fundamental limits of capacity, will be extended to channels with ISI. This will be done by defining a normalized signal-to-noise ratio,  $SNR_{\text{norm}}$ , which is a generalization of the  $SNR_{\text{norm}}$  defined in Section 8.7, with the same interpretation that  $SNR_{\text{norm}} \geq 1$  with equality if the system operates at the fundamental capacity limits.

For the remainder of this section, we will limit attention to the complex baseband channel shown in Figure 10-23c. In typical applications, this channel will have been derived from an underlying continuous-time channel by transmit and receive filtering and sampling, but we will not concern ourselves with how that happened. The ISI on the channel is represented by the transfer function  $H(e^{j2\pi fT})$ , and the additive Gaussian noise has power spectrum  $S_Z(e^{j2\pi fT})$ . When the channel is used for digital communication with PAM modulation,  $X_k$  will be replaced by the data symbols  $A_k$ , and the channel output  $Y_k$  will be applied to one of the receiver structures considered in Section 10.2, such as the LE, DFE, or MLSD. Our concern is with the performance of the latter receivers, as measured by  $P_e$ , and how that performance relates to the fundamental capacity limits for the discrete-time channel of Figure 10-23c.

Since capacity is normally expressed in bits/sec, or in terms of spectral efficiency, we need to know the sampling rate in Figure 10-23c, which is the symbol rate as well. Define the symbol rate as  $B_0 = 1/T$ , where  $T$  is the symbol interval. In typical applications, the complex baseband channel of Figure 10-23c would be associated with an underlying continuous-time passband channel. If we start with a passband channel with nominal bandwidth  $B$ , the highest possible symbol rate is  $B_0 = B$ . The complex baseband channel has bandwidth  $B/2$ , and the highest symbol rate is double this, or  $B$ .

#### Spectral Efficiency

The spectral efficiency depends on the bit rate and the bandwidth of the continuous-time channel that supports this bit rate. For the latter, we will assume that the symbol rate  $B_0$  has been chosen to be as high as possible, and thus the underlying continuous-time channel has bandwidth  $B_0$ . If the continuous-time channel actually has a higher bandwidth than this, and we are using it inefficiently by choosing a lower symbol rate than necessary, then the spectral efficiency will actually be lower than that calculated here. With this assumption, the spectral efficiency is

$$\nu_c = \frac{C}{B_0} = \frac{|F|}{B_0} \cdot \log_2 \left[ \frac{P_S / |F| + \langle S_Z / |H|^2 \rangle_{A,F}}{\langle S_Z / |H|^2 \rangle_{G,F}} \right] \text{ bits/sec-Hz.} \quad (10.135)$$

When the water-pouring band is the full bandwidth of the channel, then the factor  $|F|/B_0 = 1$ .

## Signal-to-Noise Ratio

The *signal-to-noise ratio (SNR)* is defined as the ratio of the average signal power to the total noise power. This SNR applies only to the discrete-time channel; if the underlying continuous-time channel were considered instead, the result might be different because of the bandlimiting effects of the receive filter, etc. We will now calculate the SNR when the input spectrum is chosen according to water pouring; such a spectrum can achieve capacity. There are two natural places to define the SNR; at the input to the channel, or at the output. We will focus on the input to the channel; this is simpler because the signal power at the channel output depends on the details of  $S_X$  (not just its integral) as well as the channel  $H$ . The channel-output SNR is relegated to Problem 10-14.

At the input to the channel, the signal power is, by definition,  $P_S$ . Although the channel model defines the noise spectrum as  $S_Z$  at the output of the channel  $H$ , this is equivalent to a noise with power spectrum  $S_Z/|H|^2$  at the channel input.

$$\int_F S_Z/|H|^2 df = |F| \cdot \langle S_Z/|H|^2 \rangle_{A,F}, \quad (10.136)$$

and the input SNR is then

$$SNR_{in} = \frac{P_S/|F|}{\langle S_Z/|H|^2 \rangle_{A,F}}. \quad (10.137)$$

This SNR does not depend on the water-pouring spectrum, but rather depends only on the water-pouring band  $F$  and the total signal power  $P_S$  (this is an advantage of using channel-input SNR). For the case where  $|F| = B_0$ ,  $SNR_{in}$  can also be interpreted as the signal energy per sample divided by the noise variance; in other words, in this case  $SNR_{in}$  coincides with the usual definition of SNR.

## Normalized SNR

Equation (10.135), for the maximum spectral efficiency  $B_0 = B$ , can be rewritten as

$$\frac{P_S/|F|}{2^{v_c B_0/|F|} \langle S_Z/|H|^2 \rangle_{G,F} - \langle S_Z/|H|^2 \rangle_{A,F}} = 1. \quad (10.138)$$

This is an alternative way to define the maximum achievable spectral efficiency  $v_c$  for a given set of channel parameters ( $F$ ,  $\langle S_Z/|H|^2 \rangle_{G,F}$ , and  $\langle S_Z/|H|^2 \rangle_{A,F}$ ); it is simply a generalization of (8.139) for the ideal channel. This motivates the definition of a *rate-normalized SNR*,  $SNR_{norm}$ , that is a direct generalization of (8.130). For a system operating over channel  $H$ , with output noise spectrum  $S_Z$ , with input power  $P_S$ , and achieving a spectral efficiency  $v$ , define

$$SNR_{norm} = \frac{P_S/|F|}{2^{v B_0/|F|} \langle S_Z/|H|^2 \rangle_{G,F} - \langle S_Z/|H|^2 \rangle_{A,F}} \quad (10.139)$$

This expression is simply (10.138) with the limiting spectral efficiency  $v_c$  replaced by the actual achieved spectral efficiency  $v$ . It is more complicated than it looks at first



glance, since in general  $F$  (and hence  $|F|$ ,  $\langle S_Z/|H|^2 \rangle_{A,F}$ , and  $\langle S_Z/|H|^2 \rangle_{G,F}$ ) is a complicated function of  $P_S$ . However, substituting for  $P_S$  from (10.138),

$$SNR_{\text{norm}} = \frac{2^{v_c B_0/|F|} \langle S_Z/|H|^2 \rangle_{G,F} - \langle S_Z/|H|^2 \rangle_{A,F}}{2^{v B_0/|F|} \langle S_Z/|H|^2 \rangle_{G,F} - \langle S_Z/|H|^2 \rangle_{A,F}} \geq 1, \quad (10.140)$$

with equality if the system is operating at the fundamental limits of capacity. That is, the relation  $v \leq v_c$  is equivalent to the bound  $SNR_{\text{norm}} \geq 1$ . Note that (10.140) does not require that the system occupy the water-pouring band  $F$ , but rather applies to any system operating with transmit power  $P_S$  and spectral efficiency  $v$ .

We can express  $SNR_{\text{norm}}$  in terms of the  $SNR_{\text{in}}$  by substituting for  $P_S$  from (10.137),

$$SNR_{\text{norm}} = \frac{SNR_{\text{in}}}{2^{v B_0/|F|} \langle S_Z/|H|^2 \rangle_{G,F} / \langle S_Z/|H|^2 \rangle_{A,F} - 1}, \quad (10.141)$$

a relation that is similar to, and a generalization of, the ideal channel case in (8.130).

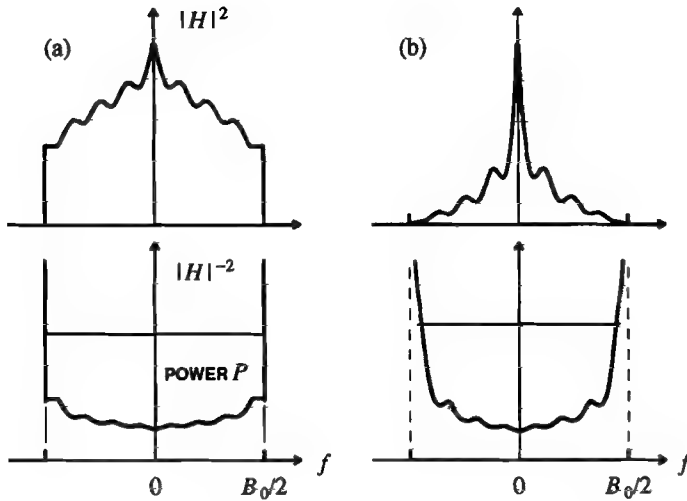
### Example 10-27.

For a complex baseband discrete-time channel derived from a continuous-time ideal passband channel with bandwidth  $B$  and white noise  $N_0$ ,  $S_Z = 2N_0$  and  $H = 1$ , and the water-pouring band will be the full bandwidth,  $F = [-B/2, B/2]$ . Hence the arithmetic and geometric means will both be  $2N_0$ , and  $SNR_{\text{norm}} = SNR_{\text{in}}/(2^v - 1)$ . Furthermore,  $SNR_{\text{in}} = P_S/2N_0B$ , and this expression for  $SNR_{\text{norm}}$  is the same definition as (8.130). Unlike that simpler expression, (10.141) is a function of the ISI on the channel through the parameters  $F$ ,  $\langle S_Z/|H|^2 \rangle_{G,F}$ , and  $\langle S_Z/|H|^2 \rangle_{A,F}$ .  $\square$

It is important to note that in  $SNR_{\text{norm}}$ ,  $F$  depends on  $P_S$  (through water pouring), and hence the other factors ( $|F|$ ,  $\langle S_Z/|H|^2 \rangle_{A,F}$ , and  $\langle S_Z/|H|^2 \rangle_{G,F}$ ) also depend on  $P_S$ . Thus, unlike in the ideal channel case,  $SNR_{\text{norm}}$  is in general not a linear function of  $P_S$ . As in Section 8.7,  $10\text{-log}_{10} SNR_{\text{norm}}$  expressed in dB is related to the "SNR gap to capacity". In Section 8.7, that SNR gap to capacity was defined as the increase in transmitted power (or equivalently  $SNR$ ) required to achieve a give  $P_e$ , relative to the Shannon limit on power at the same spectral efficiency. Unfortunately that same simple interpretation does not apply to the ISI case considered here, because  $SNR_{\text{norm}}$  is not linearly related to the transmit power  $P_S$ . However, based on  $SNR_{\text{norm}}$  we could still infer the SNR gap to capacity by taking into account the precise non-linear relationship between  $P_S$  and  $SNR_{\text{norm}}$ . Fortunately, at high  $SNR$  this is not necessary, and the simpler interpretation of Section 8.7 applies, as we now show.

### Capacity at High SNR

On some (but not all) channels  $H$  the water-pouring bandwidth eventually becomes the full bandwidth  $|F| = B_0$  at high signal powers. Two cases are illustrated in Figure 10-26 (for the white noise case). If  $|H|^2$  is non-zero for all  $|f| \leq B_0$  as in Figure 10-26a, the sides of the bowl become infinitely steep at the band edge, and for sufficiently high signal powers  $|F| = B_0$ . In contrast, on a channel with a zero at the



**Figure 10-26.** Illustration of water pouring on channels with and without zeros. (a) When  $H$  is non-zero for all  $|f| \leq B_0$ , the bowl has infinitely steep sides at the band edge. (b) When  $H$  has a zero at the band edge, the sides of the bowl have finite slope, and water pouring never fills the bandwidth. Similar behavior will occur if  $H$  has a zero anywhere in-band.

band edge, as shown in Figure 10-26b, or for that matter a zero anywhere else,  $|f| < B_0$  at all finite (even if large) signal powers.

If  $P_S$  ever becomes large enough that  $|F| = B_0$ , a major simplification occurs, in that the parameters  $\langle S_Z/|H|^2 \rangle_{A,F}$  and  $\langle S_Z/|H|^2 \rangle_{G,F}$  are constants that are not a function of  $P_S$  and  $SNR_{in}$  is proportional to  $P_S$ . In that case,  $SNR_{norm}$  has the same simple interpretation as that in Section 8.7; namely, it is the increase in signal power (or equivalently  $SNR_{in}$ ) required to achieve a given  $P_e$ , relative to the Shannon limit.

#### Example 10-28.

If the noise on a passband channel is white,  $S_Z = 2N_0$ , and  $|F| = B_0$ , then from (10.139)

$$SNR_{norm} = \frac{P_S / 2N_0 B_0}{2^v \cdot \langle |H|^{-2} \rangle_G - \langle |H|^{-2} \rangle_A} \quad (10.142)$$

The only term on the right that is a function of  $P_S$  is  $P_S$  itself. Thus,  $SNR_{norm}$  and  $P_S$  are directly proportional (with the domain of applicability of (10.142)).  $\square$

#### Example 10-29.

In both Figure 10-24 and Figure 10-25, the high SNR approximation, obtained by setting  $|F| = B_0$  and replacing the arithmetic and geometric means by their values over the full Nyquist bandwidth, are plotted. These asymptotes are useful at high SNR, especially in the single-pole channel case. However, the single-zero case illustrates that very high signal powers may be needed to approach the asymptote, because  $H$  will be very small in the vicinity of a zero near the unit circle. For the case of  $|c| = 1$ , where there is a null in  $H$  at the angle of  $c$ , the arithmetic mean will not even exist and the water-pouring band will never

fill the Nyquist interval. In this case, there is no meaningful high-power asymptote.  $\square$

Whenever a channel has nulls within its bandwidth, the water-pouring band  $F$  will always exclude a small frequency interval around these nulls. This is fortunate, because otherwise  $\langle S_Z/|H|^2 \rangle_{A,F}$  would be infinite! If  $|H|^{-2}$  is zero over an interval of frequencies, then  $F$  will exclude that interval, which is fortunate since otherwise  $\langle S_Z/|H|^2 \rangle_{G,F}$  and  $\langle S_Z/|H|^2 \rangle_{A,F}$  would both be infinite. In fact, the water-pouring procedure ensures that  $|H|^2 > 0$  over  $F$ , and therefore both  $\langle S_Z/|H|^2 \rangle_{G,F}$  and  $\langle S_Z/|H|^2 \rangle_{A,F}$  are bounded for all finite powers  $P_S$ .

However, nulls in  $|H|^2$  do imply that  $|F| < B_0$  for all  $P_S$ . Thus, the simple interpretation of  $SNR_{\text{norm}}$  never occurs, because  $\langle S_Z/|H|^2 \rangle_{G,F}$  and  $\langle S_Z/|H|^2 \rangle_{A,F}$  are functions of  $P_S$  at all power levels and  $SNR$  never becomes precisely proportional to  $P_S$ . On the other hand, for simple isolated nulls, the interval excluded from the water-pouring band will shrink to zero, and we would not expect the nulls to have a large effect at high  $P_S$ . This issue has to be addressed carefully, and would take us too far afield here [17,18].

Remarkably, at high SNR, provided that  $|F| = B_0$ , the capacity of the discrete-time channel is directly related to the MSE's of the LE-ZF and DFE-ZF given by (10.61) and (10.84),

$$C = B_0 \cdot \log_2 \left\{ \frac{\epsilon_{\text{LE-ZF}}^2 + P_S/B_0}{\epsilon_{\text{DFE-ZF}}^2} \right\} \quad \text{bits/sec} . \quad (10.143)$$

At high SNR, the capacity of a discrete-time channel can be predicted by knowing  $\epsilon_{\text{LE-ZF}}^2$  and  $\epsilon_{\text{DFE-ZF}}^2$  alone. The normalized SNR becomes

$$SNR_{\text{norm}} = \frac{P_S/B_0}{2^v \cdot \epsilon_{\text{DFE-ZF}}^2 - \epsilon_{\text{LE-ZF}}^2} . \quad (10.144)$$

### 10.5.5. Relationship of Capacity to Unbiased DFE-MSE

The formula for capacity of (10.143), which applies to many channels at high signal powers, can be replaced by an even simpler formula,

$$C = B_0 \cdot \log_2 (1 + SNR_{\text{MSE-DFE-U}}) , \quad SNR_{\text{MSE-DFE-U}} = \frac{\sigma_A^2}{\epsilon_{\text{DFE-MSE-U}}^2} , \quad (10.145)$$

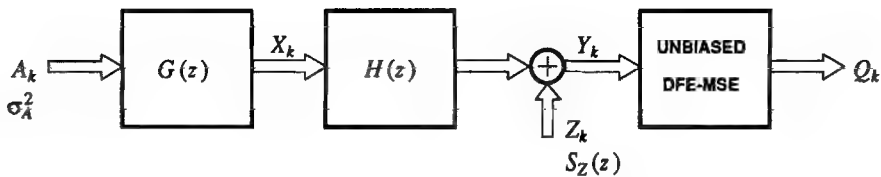
where  $\epsilon_{\text{DFE-MSE-U}}^2$  is the MSE of the unbiased DFE-MSE designed in Section 10.2. Remarkably, this formula holds at *all* signal powers, whereas (10.143) holds only at high power. To understand this result, the design of the unbiased DFE-MSE that leads to (10.145) needs to be clarified:

- Unlike the ZF case, the statistics of the data symbols matter to the MSE criterion, so they must be specified. The input symbols  $A_k$  are chosen to be zero-mean, stationary, and mutually independent (and hence white).

- A discrete-time transmit filter  $G(z)$  is added, filtering the data symbols before they reach the channel, in order to control the shape of the spectrum of the signal at the discrete-time channel input subject to an average power constraint. (Note that the underlying continuous-time channel will normally also include a continuous-time transmit filter, which is not affected.)
- The unbiased MSE between the equalizer output and the data symbols is minimized by choice of the receiver precursor and postcursor filters, taking into account  $G(z)$ .
- The MSE is minimized by the choice of  $G(z)$ , which remarkably turns out to result in a channel-input spectrum exactly to the water-pouring spectrum that achieves capacity.

This result demonstrates a close canonical relationship between the DFE-MSE-U and channel capacity. In particular, (10.145) is in precisely the functional form of the capacity for an ideal white noise channel, except that the SNR of the DFE-MSE-U is substituted for the SNR of the ideal channel. The optimal  $G(z)$  for the DFE-MSE-U results in a transmit spectrum reaching the channel that is precisely the water-pouring spectrum for that channel. Furthermore, (10.145) holds at all SNR, and is not just an asymptotic result. Thus, the SNR of the DFE-MSE-U, for an optimized transmit filter that results in the water-pouring spectrum at the channel input, bears the same relationship to capacity on channels with ISI as the simple SNR plays in the absence of ISI. This result was recently derived [9] in a more general continuous-time channel context. The derivation we give here is simpler but also less general.

The derivation of (10.145) is based on the configuration of Figure 10-27. The input data symbols  $A_k$  are zero-mean and white with variance  $\sigma_A^2$ . Furthermore, we assume that they are mutually independent. The data symbols are put through  $G(z)$ , which allows us to generate any input power spectrum to the channel  $\sigma_A^2 |G_h|^2$ . Since the effects of channel noise could be eliminated by choosing a transmit filter with a very large gain, we constrain the power at the transmit filter output to be  $P_S$ , or in other words set  $P_S \cdot T = \sigma_A^2 < |G|^2 >_A$ . A DFE-MSE-U equalizer is put at the output of the channel, generating output samples  $Q_k$ , which would then be applied to a slicer.



**Figure 10-27.** The configuration used for deriving the canonical relationship between the DFE-MSE-U and channel capacity.

The earlier analysis of Section 10.2 considered a similar situation, and determined the MSE assuming the data symbols are white and independent. That analysis did not include any discrete-time transmit filter  $G(z)$ , but that shortcoming is trivially overcome by recognizing that the transmit filter can simply be combined with the channel  $H$ , and those earlier results can be applied with  $H$  replaced by  $GH$ . We first minimize the MSE for any given transmit filter  $G$ , and subsequently choose  $G$  to minimize the MSE under the transmit power constraint. From (10.95) and (10.84),

$$\frac{\sigma_A^2}{\epsilon_{\text{DFE-MSE-U}}^2} + 1 = \frac{\sigma_A^2}{\epsilon_{\text{DFE-MSE}}^2} = \frac{A_y^2}{A_z^2} = \frac{\langle S_Y \rangle_G}{\langle S_Z \rangle_G} = \langle S_Y/S_Z \rangle_G, \quad (10.146)$$

where

$$S_Y = \sigma_A^2 |GH|^2 + S_Z. \quad (10.147)$$

If we define  $S_X = \sigma_A^2 |G|^2$  as the input power spectrum to the channel, we get that

$$S_Y = S_X |H|^2 + S_Z, \quad (10.148)$$

and

$$\frac{\sigma_A^2}{\epsilon_{\text{DFE-MSE-U}}^2} + 1 = \langle 1 + S_X |H|^2 / S_Z \rangle_G. \quad (10.149)$$

This gives us the minimum MSE for any given transmit filter  $G$ . The final step is to choose  $G$  to minimize the MSE, subject to the constraint that the power in  $S_X$  is  $P_S$ . We can relate this optimization back to capacity, since from (10.123),

$$2^{C/B_0} = \langle 1 + S_X |H|^2 / S_Z \rangle_G. \quad (10.150)$$

The right side of (10.150) is maximized when  $S_X$  is the water-pouring spectrum. Thus, maximizing the SNR of (10.149) is mathematically equivalent to finding the capacity, and the transmit spectrum  $S_X$  that maximizes (10.149) is the water-pouring spectrum. When  $S_X$  is determined by water pouring, we can set (10.149) and (10.150) equal,

$$\frac{\sigma_A^2}{\epsilon_{\text{DFE-MSE-U}}^2} + 1 = 2^{C/B_0}, \quad (10.151)$$

thereby establishing (10.145).

### 10.5.6. Impact of ISI on Capacity

The effect of channel  $H$  and noise  $S_Z$  on the capacity can be quantified by setting the capacity of the channel with ISI equal to the capacity of an ideal channel and solving for the relationship between the transmitted power required in both cases. This comparison is most meaningful if we assume that the noise is white in both cases,  $S_Z = 2N_0$ , where we get

$$\frac{P_{S,\text{isi}}}{2N_0|F|} = \langle |H|^{-2} \rangle_{G,F} \cdot \left[ \frac{P_{S,\text{ideal}}}{2N_0B_0} + 1 \right]^{B_0/|F|} - \langle |H|^{-2} \rangle_{A,F}. \quad (10.152)$$

$P_{S, \text{isi}}$  can be larger or smaller than  $P_{S, \text{ideal}}$ . For example, if the channel has a large gain, the channel with ISI may require less transmitted power in spite of the ISI. To separate out the effects of ISI, we can normalize the channel response, for example by setting the received isolated pulse energy  $\sigma_h^2$  equal to that of the ideal channel.

As the transmitted power increases,  $|F| \rightarrow B_0$ , and (10.152) simplifies to

$$SNR_{\text{in, isi}} = \langle |H|^{-2} \rangle_G \cdot (SNR_{\text{in, ideal}} + 1) - \langle |H|^{-2} \rangle_A. \quad (10.153)$$

Asymptotically, at high SNR, the effect of the ISI is to increase the required  $SNR_{\text{in}}$  by a factor of  $\langle |H|^{-2} \rangle_G$ .

### Example 10-30.

Given a minimum-phase causal IIR channel with unit-energy impulse response and a single pole,

$$H(z) = \frac{\sqrt{1 - |c|^2}}{1 - cz^{-1}} \quad (10.154)$$

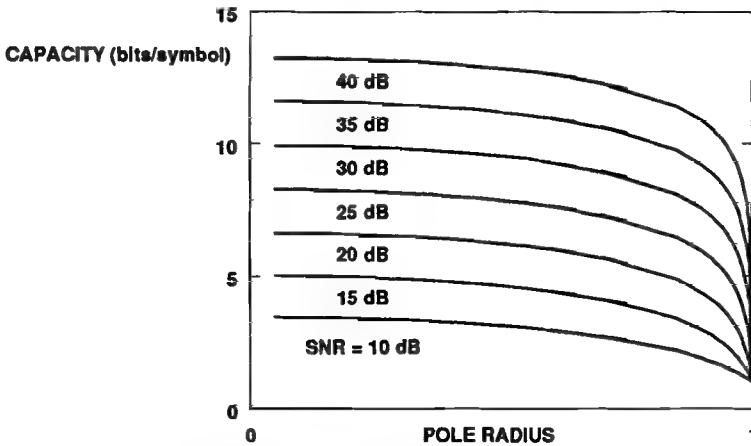
for  $|c| < 1$ , the means are

$$\langle |H|^{-2} \rangle_A = \frac{1 + |c|^2}{1 - |c|^2}, \quad \langle |H|^{-2} \rangle_G = \frac{1}{1 - |c|^2}. \quad (10.155)$$

Thus, the capacity at high SNR is

$$C \cdot T = \log_2 (SNR \cdot (1 - |c|^2) + (1 + |c|^2)), \quad (10.156)$$

where  $SNR = P_S / 2N_0 B_0$ . This formula can be used for conveniently illustrating the effect of pole radius on capacity, as shown in Figure 10-28. As  $|c| \rightarrow 1$ , the ISI gets worse and there is a rapid drop-off in capacity. Asymptotically at high SNR, the channel with ISI



**Figure 10-28.** The capacity per symbol (spectral efficiency) of a one-pole channel (with impulse response energy normalized to unity) plotted as a function of pole radius.

requires a larger SNR by a factor of  $1/(1 - |c|^2)$  to achieve the same capacity as the ideal channel ( $c = 0$ ). For example, in the case plotted in Figure 10-25,  $|c| = 0.99$ , and the SNR must be larger by 20 dB, which is the asymptotic difference between the "ideal channel" and "actual channel" curves in Figure 10-25.  $\square$

### 10.5.7. Performance of the LE and DFE

Using the results from the previous section, the performance of the linear and decision-feedback equalizers is easily related to capacity limits by calculating  $SNR_{\text{norm}}$  at a given error probability, and comparing to unity. This comparison is simple at high signal power, if the water pouring bandwidth becomes the full bandwidth of the channel, so we will restrict ourselves to this case. Generally this corresponds to the case where both the LE-ZF and DFE-ZF exist, further simplifying matters.

For a zero-forcing equalizer, if the transmit symbol is  $\alpha A_k$  (the factor  $\alpha$  allows us to vary the transmit power), then the input to the slicer is  $\alpha A_k$  plus additive Gaussian noise with variance  $\epsilon^2$  for MSE  $\epsilon^2$  (where  $\epsilon^2 = \epsilon_{\text{LE-ZF}}^2$  or  $\epsilon^2 = \epsilon_{\text{DFE-ZF}}^2$ ). Since the Gaussian noise is circularly symmetric, the real and imaginary parts have equal variance  $\sigma^2 = \epsilon^2/2$ . In the DFE-ZF case, this presumes optimistically that the slicer has made no past decision errors within the memory of the postcursor equalizer. The minimum distance is  $\alpha a_{\min}$ , where  $a_{\min}$  is the minimum distance over the symbol alphabet. When the data symbols are independent and each have variance  $\sigma_A^2$ , the transmitted energy per symbol is  $P_S T = \alpha^2 \sigma_A^2$ . If  $K$  is the average number of symbols at the minimum distance, the error probability is approximately

$$P_e \approx K \cdot Q\left(\frac{d_{\min}}{2\sigma}\right) = K \cdot Q\left[\sqrt{\frac{\alpha^2 a_{\min}^2}{2\epsilon^2}}\right] = K \cdot Q\left[\sqrt{\frac{a_{\min}^2 P_S T}{2\sigma_A^2 \epsilon^2}}\right]. \quad (10.157)$$

This gives us an accurate estimate of the error probability for zero-forcing equalizers, but we are interested in their performance in relation to capacity. If the PAM system operates at minimum bandwidth (zero excess bandwidth), then the spectral efficiency is  $\nu = \log_2 M$ . Furthermore, if the input noise is white, at sufficiently high power levels the water-pouring band becomes the entire Nyquist interval, and the normalized signal-to-noise ratio becomes, from (10.144), generalized to a complex baseband channel,

$$SNR_{\text{norm}} = \frac{P_S T}{2^{\nu} \epsilon_{\text{DFE-ZF}}^2 - \epsilon_{\text{LE-ZF}}^2}, \quad (10.158)$$

and substituting for  $P_S T$  in (10.157),

$$P_e \approx K \cdot Q(\sqrt{\gamma_A \gamma_{\text{isi}} SNR_{\text{norm}}}), \quad (10.159)$$

where

$$\gamma_A = \frac{(M-1)a_{\min}^2}{2\sigma_A^2} \quad (10.160)$$

is a property of the symbol constellation (already defined for the ideal channel case in

Section 8.7, (8.143)), and

$$\gamma_{\text{isi}} = \frac{M \epsilon_{\text{DFE-ZF}}^2 - \epsilon_{\text{LE-ZF}}^2}{(M-1)\epsilon^2} \quad (10.161)$$

is a function of the choice of equalizer structure (through  $\epsilon^2$ ) and also the size of the constellation  $M$ . We saw in Section 8.7 that for square QAM constellations,  $\gamma_A = 3$  independent of  $M$ , the size of the constellation. Thus, the effect of ISI on the SNR gap to capacity is embodied in the term  $\gamma_{\text{isi}}$ , which is also a function of  $M$ , the constellation size.

Equation (10.159) is a remarkably simple characterization of  $\text{SNR}_{\text{norm}}$  that applies at high SNR to discrete-time channels  $H$  with additive white noise  $S_Z = 2N_0$ . It reflects the nature of the ISI on the channel through only two parameters, the MSE of the LE-ZF and DFE-ZF. To reiterate, it applies only when the water-pouring bandwidth fills the entire Nyquist bandwidth. The effect of the ISI is reflected in the factor  $\gamma_{\text{isi}}$ , which will generally be less than unity, requiring  $\text{SNR}_{\text{norm}}$  to be larger to achieve the same error probability. Thus, the difference between  $\gamma_{\text{isi}}$  and unity is a measure of the increase in the SNR gap to capacity due to ISI, for the particular equalizer structure used. Of course, when there is no ISI,  $\epsilon_{\text{DFE-ZF}}^2 = \epsilon_{\text{LE-ZF}}^2 = \epsilon^2$ , and  $\gamma_{\text{isi}} = 1$  and the SNR gap to capacity reduces to that displayed for the ideal channel in Section 8.7.

For large signal constellations  $M$  (large  $v$ ),  $\gamma_{\text{isi}}$  approaches an asymptote

$$\gamma_{\text{isi}} \rightarrow \frac{\epsilon_{\text{DFE-ZF}}^2}{\epsilon^2}. \quad (10.162)$$

There are two cases of interest:

- When we use the LE-ZF, then for large  $M$

$$\gamma_{\text{isi}} \rightarrow \frac{\epsilon_{\text{DFE-ZF}}^2}{\epsilon_{\text{LE-ZF}}^2}, \quad (10.163)$$

and the SNR gap to capacity is increased, relative to the ideal channel case, by precisely the MSE penalty of the LE-ZF relative to the DFE-ZF.

- If we use the DFE-ZF, then for large  $M$

$$\gamma_{\text{isi}} \rightarrow 1, \quad (10.164)$$

and there is no increase in the SNR gap to capacity due to ISI.

This last fact is a remarkable conclusion first observed by Price [19]. It says that the SNR gap to capacity at high signal powers is independent of the ISI, and in particular is the same for channels with ISI and for ideal channels, as long as the DFE-ZF is used as the receiver structure. Thus, channel coding has potentially the same benefit (the same gap to close) on channels with ISI as on ideal channels. This statement has to be qualified, however, with many caveats:

- The water-pouring band must fill the entire baseband bandwidth at high signal powers; that is, the channel transfer function must be non-zero. (However, it has



been shown that the result is unchanged if the channel has isolated zeros, for example at the band edge.)

- While this statement is true for general symbol-rate sampled discrete-time channels, it applies to continuous-time channels only if they can be represented by an equivalent discrete-time channel through transmit/receive filtering and sampling without information loss. That is, they must be *strictly bandlimited*. (It has been shown that this asymptotic result does not apply to many channels that are not strictly bandlimited, as will be discussed later.)
- It applies asymptotically only at high signal powers and large signal constellations.
- Our approximation to error probability assumes that all past decisions are correct. In fact, the error probability of the DFE-ZF will be higher, due to error propagation. As shown in Appendix 10-A, the effect on error probability is to increase the factor  $K$ , so it is not a major effect.

Price's result suggests that, at least in principle, channel coding techniques designed for the ideal Gaussian channel should have approximately the same potential benefit (reduction in signal power for the same  $P_e$ ) on channels with ISI, independent of the nature of the ISI, at high signal powers. This follows from the fact that the SNR gap to capacity for the DFE-ZF is independent of the nature of the ISI, including the case of no ISI, at high signal powers. This result also suggests that the DFE-ZF is a good receiver structure on which to base a channel coding system for channels with ISI, since channel coding techniques should be able to close the SNR gap to capacity to the same extent from a starting point of the DFE-ZF as on the ideal channel. In particular, it shows that the MLSD receiver, which is optimal with respect to probability of error in the absence of channel coding, is not required at high signal powers in the presence of channel coding to obtain good performance. This result is surprising in light of the fact that the DFE-ZF arbitrarily cancels a part of the received signal energy, the part residing in the postcursor ISI, while the MLSD uses this energy to advantage.

There are some obstacles to applying Price's result in practice, since it depends on the ability to cancel the postcursor ISI using past decisions. This depends on the constellation being discrete, and further depends on immediate decisions, both of which are incompatible with available channel coding techniques. Fortunately, ways have been found to circumvent these obstacles, as discussed in Chapter 14.

It is tempting to apply a similar analysis to the DFE-MSE-U. Since it is canonically related to capacity, we might expect its SNR gap to capacity to be independent of ISI at all signal powers (as long as the transmit filter is designed in accordance with water pouring). However, we cannot estimate the error probability of the DFE-MSE-U simply, since there is residual ISI at the slicer (and hence the slicer error is not Gaussian); further, the Gaussian noise component of the slicer error is not white. Thus, the extent to which the DFE-MSE-U may or may not close the SNR gap to capacity at low signal powers, relative to the DFE-ZF, remains an open question.

## 10.6. FURTHER READING

For further details on the design of optimal detectors in various circumstances arising in digital communication, see Wozencraft and Jacobs [20] or Proakis [21]. An excellent treatise on equalization can be found in [22] or [15]. The literature on the subject is very extensive, and those references have extensive bibliographies. On the Viterbi algorithm for sequence detection, the original paper by Forney [23] and a later tutorial article [24] are highly recommended. An excellent tutorial description of transmitter precoding, Price's result, and the capacity of channels with ISI is [25]. A nonlinear equalizer structure using decision-aided cancellation not covered here is the *intersymbol interference canceler* suggested by Proakis [26] and elaborated by Gersho and Lim [27]. Simplifications of the full-blown MLSD for ISI channels have been explored in several articles [28,29,30].

### APPENDIX 10-A DFE ERROR PROPAGATION

In this appendix we consider the nature of error propagation in the DFE, finding in particular upper bounds on the error probability that demonstrate that the effects are usually insignificant compared to the benefits of reduced noise enhancement.

We can model the error propagation phenomenon by removing the assumption that  $A_k = \hat{A}_k$ , and rewriting the slicer input as

$$Q_k = A_k + \sum_{j=1}^{\infty} g_j A_{k-j} - \sum_{j=1}^{\infty} g_j \hat{A}_{k-j} + Z_k \quad (10.165)$$

where  $Z_k$  is the complex-valued noise at the slicer input. This can be rewritten as

$$Q_k = A_k + V_k + Z_k \quad (10.166)$$

where the middle term is the residual ISI due to incorrect cancellation of postcursor ISI samples given by

$$V_k = \sum_{j=1}^N g_j W_{k-j}, \quad W_k = A_k - \hat{A}_k, \quad (10.167)$$

and a finite number  $N$  taps has been assumed.

For purposes of understanding this phenomenon further, specialize to the baseband case with binary antipodal signaling, so that  $A_k = \pm 1$  and  $W_k$  assumes the values  $\{\pm 2, 0\}$ . For this case the slicer will apply a threshold at zero, and we can easily calculate the probability for both types of error at time  $k$ . The first type of error occurs when  $A_k = 1$  and the slicer input is negative, and results in  $W_k = 2$ ,

$$\Pr\{W_k = 2\} = p \cdot \Pr\{1 + V_k + Z_k < 0\} \quad (10.168)$$

where  $p = \Pr\{A_k = 1\}$ . Similarly the probability that  $W_k = -2$  is

$$\Pr\{W_k = -2\} = (1-p) \cdot \Pr\{-1 + V_k + Z_k > 0\} \quad (10.169)$$

**Exercise 10-6.**

Show that if  $Z_k$  is a real-valued zero-mean Gaussian random variable, and the data symbols are equally likely,  $p = 1/2$ , then for any ISI  $V_k$

$$\Pr\{W_k \neq 0\} < 1/2. \quad (10.170)$$

□

The intuition behind this result is that no matter how big the ISI, it actually *reduces* the error probability for one polarity of data symbol, and if the symbols are equally probable then the error probability can be no worse than  $1/2$  for any residual ISI.

Of course the conclusion that the error probability is no worse than  $1/2$  is of little value since we could flip a coin in the receiver and do just as well. In order to get stronger results, assume that the data symbols  $A_k$  and the noise samples  $Z_k$  are independent. From Section 10.6 we know that the noise samples will be independent as  $N \rightarrow \infty$  for the optimal forward filter design, and should be approximately true for  $N$  sufficiently large. Then we can see that  $\Pr\{W_k\}$  depends only on  $W_{k-j}$ ,  $1 \leq j \leq N$ , and therefore  $W_k$  is a Markov chain (Section 3.3) with  $3^N$  states. While the steady-state probability of the states of this chain can be calculated in principle [2], we can easily develop a simple model for this chain that gives an upper bound on the error probability as well as considerable insight into the error propagation phenomenon [31].

Assume that in the absence of ISI the error probability is  $P_{e,0}$ ,

$$P_{e,0} = \Pr\{W_k \neq 0 | W_{k-j} = 0, 1 \leq j \leq N\}. \quad (10.171)$$

Further, make the worst case assumption that if there is residual ISI, the error probability is  $1/2$ ,

$$\Pr\{W_k \neq 0 | W_{k-j} \neq 0 \text{ for some } j, 1 \leq j \leq N\} = 1/2. \quad (10.172)$$

Define a Markov chain  $X_k$ , where  $X_k$  is a count of the number of successive correct decisions that have been made up to but not including time  $k$ . That is,  $X_k = n$  if  $W_{k-1} = W_{k-2} = \dots = W_{k-n} = 0$  and  $W_{k-n-1} \neq 0$ . We now get the following model for an error propagation event. If  $X_k \geq N$  there is no residual interference because there have been no errors made within the memory of the FIR DFE feedback filter. Assume that  $X_k \geq N$  but that an error is made anyway at time  $k$ ,  $W_k \neq 0$ , due to the additive noise, resulting in  $X_{k+1} = 0$ . Then according to our worst-case model, errors will be made thereafter with probability  $1/2$  until such time that  $N$  correct decisions in a row have been made, at which time we revert to the state of zero residual ISI and the error probability returns to  $P_{e,0}$ . Suppose that on average it takes  $K$  time increments until  $N$  correct decisions in a row have been made. We call  $K$  the average length of an error propagation event. Since the error probability is  $1/2$  during the event, error propagation results in an average of  $1/2 \cdot K$  errors for every error due to the random noise. Thus, the error probability taking into account ISI is

$$P_e = (1/2 \cdot K + 1) P_{e,0} . \quad (10.173)$$

In actuality we expect that the error probability will be less than this, since the error probability during an error event will in fact be less than  $1/2$ . This bound can be strengthened and made more rigorous as in [31].

This logic, even though worst-case, gives considerable insight into the mechanism of error propagation. It demonstrates that error events will terminate whenever we make  $N$  correct decisions in a row, and that this happens in relatively short order because the error probability is no worse than  $1/2$  no matter how large the ISI gets. This argument depends strongly on the assumption of equally probable (random) data, and in fact there are worst-case data sequences for which the error event can persist much longer than we have predicted here. This suggests that it is important to insure that the data is random when the DFE is used, and also suggests that it is desirable to keep  $N$  as small as possible consistent with obtaining most of the benefit of the DFE.

We can determine  $K$  in (10.173) by observing that  $K$  is the average number of tosses of a fair coin before we obtain  $N$  heads in a row. This was determined in Problem 3-13 to be

$$K = \frac{1 - 1/2^N}{1/2^{N+1}} = 2(2^N - 1) . \quad (10.174)$$

Substituting into (10.173), we get

$$P_e = 2^N \cdot P_{e,0} \quad (10.175)$$

and the error probability is multiplied by a factor of  $2^N$  due to error propagation. If  $N$  is fairly modest, this error multiplication is much more than offset by the benefit of the DFE in reducing  $P_{e,0}$ . For example,  $N = 3$  results in an order of magnitude increase in error probability due to error propagation, and the DFE must reduce  $P_{e,0}$  by only about half a dB to result in a net reduction in error probability.

## PROBLEMS

- 10-1. Extend Example 10-4 to a channel with two zeros, and the same alphabet.
- Sketch one stage of the trellis, showing only the structure of the trellis (allowable branches) and not bothering to label the branch metrics.
  - Sketch the error event corresponding to a single error; that is, sketch the one which, if it is the minimum-distance error event, implies that  $\gamma_{\text{MLSD}} = \gamma_{\text{MF}}$ .
  - Sketch two other error events which are relatively short, and are thus possible candidates as the minimum-distance error event.
- 10-2. Show that the inequality in (10.28) is strict when  $G_h(z)$  is FIR and  $G_h(z) \neq 1$ . **Hint:** You will need to use the fact that only a finite number of error events need be considered in calculating the minimum distance.
- 10-3. Given a channel with transfer function  $H(z) = 1/(1 - cz^{-1})$ , with complex-valued pole location  $c$  such that  $|c| \neq 1$ , verify the following:
- $\langle |H|^{-2} \rangle_A = 1 + |c|^2$

- (b)  $\langle |H|^{-2} \rangle_G = 1$  for  $|c| < 1$  and  $\langle |H|^{-2} \rangle_G = |c|^2$  for  $|c| > 1$ .
- (c) Note that the geometric mean is everywhere smaller, as expected.
- 10-4.** Consider the channel of Problem 10-3:
- (a) Find the transfer function of the LE-ZF, assuming the noise is white.
- (b) Find  $\epsilon_{\text{LE-ZF}}^2$  for both the minimum-phase and maximum-phase cases.
- (c) Interpret the result as a function of  $|c|$ .
- (d) Discuss the practicality of this channel model.
- 10-5.** Repeat Problem 10-4, except normalize the channel impulse response such that the energy in the impulse response is unity for all  $c$ . Be sure to treat the minimum- and maximum-phase cases separately, and explain the results intuitively.
- 10-6.** For the channel model of Problem 10-3 and white noise:
- (a) When  $|c| < 1$ , find the precursor and postcursor equalizers and the resulting MSE.
- (b) Repeat (a) for  $|c| > 1$ .
- (c) Interpret the results of (a) and (b), and compare to the results of Problem 10-4.
- 10-7.** As in Problem 10-5, modify the results of Problem 10-6 by normalizing the channel impulse response to unit energy.
- 10-8.** Show that  $\epsilon_{\text{DFE-MSE}}^2$  of (10.84) approaches  $\epsilon_{\text{DFE-ZF}}^2$  as  $S_Z \rightarrow 0$ , and further that the precursor and postcursor equalizers for the MSE criterion approach those of the DFE-ZF.
- 10-9.** Derive the structure of a FSE for the following circumstances, using a sampling rate that is no higher than necessary:
- (a) The received pulse has excess bandwidth less than 200%.
- (b) The received pulse has excess bandwidth less than 50%.
- 10-10.** Assuming that the FSE of Figure 10-18c is implemented as a non-FIR discrete-time filter, derive a time-domain expression for the output  $Q_k$  in terms of the samples  $X_k$  at the output of the anti-aliasing lowpass filter  $F(j\omega)$ .
- 10-11.** Derive a formula for the capacity of the continuous-time passband channel in terms of arithmetic and geometric means for an equivalent complex baseband channel model.
- 10-12.** Suppose the method for deriving a discrete-time baseband channel from a continuous-time baseband channel of Figure 10-22 is replaced by a more practical system, where the transmit filter is replaced by a general filter  $\sqrt{T}G(j2\pi f)$ , and the receive filter is replaced by a general filter  $\sqrt{T}F(j2\pi f)$ . Also do not assume that the filters  $G$ ,  $H$ , or  $F$  are necessarily bandlimited.
- (a) Derive a formula for the capacity of the resulting discrete-time channel.
- (b) Derive the capacity when the receive filter is chosen to be a filter matched to the received pulse.
- (c) Under what conditions can you be certain that the capacity of the discrete-time channel of (a) has the same capacity as the underlying continuous-time channel?
- (d) Repeat (c) for the receive filter of (b).
- (e) Repeat (d) when a discrete-time precursor equalizer is added to turn the receiver front end into a WMF.
- 10-13.** Show that the discrete-time capacity formulas for the baseband and passband cases both reduce to the capacity of (10.111) when the noise is white and the discrete-time channel is ideal.
- 10-14.** In the chapter, the normalized SNR was expressed in terms of the channel-input SNR. In this problem, we will reformulate it in terms of the channel-output SNR.
- (a) Derive an expression for the SNR at the output of the channel within the water-pouring band,  $\text{SNR}_{\text{out}}$ , where the signal power is derived for the water-pouring signal spectrum.

- (b) Utilizing the result of (a), express  $SNR_{\text{nom}}$  in terms of  $SNR_{\text{out}}$ .
- 10-15. With the DFE-MSE-U, we found benefit in including a transmit filter, and optimizing that filter resulted in the water-pouring filter at the channel input. The question arises as to the benefit of a transmit filter to the DFE-ZF. Include a transmit filter  $G$  in the DFE-ZF, assuming the data symbols are white ( $G_a = 1$ ) and the transmit power is constrained to  $P_S$ ,  $P_S \cdot T = \sigma_A^2 < |G|^2 >_A$ . Under these conditions, prove that the optimal transmit filter, with the goal of minimizing the MSE, is a flat filter. That is, the trivial transmit filter is optimal.
- 10-16. Generalize the results of Problem 10-15 by allowing the transmit data symbols to have a general power spectrum  $S_A$ .
- (a) Show that the optimal transmit filter for the DFE-ZF is not flat, but is in fact a whitening filter for the transmit symbols.
- (b) Find the resulting MSE.
- (c) Show that the MSE is always smaller when the transmit symbols are not white and the optimal transmit filter is used, than when the symbols are white with the same variance.
- 10-17. Since  $\epsilon_{\text{DFE-MSE}}^2 \leq \epsilon_{\text{DFE-ZF}}^2$ , it might appear from (10.162) that a DFE-MSE could potentially operate at a smaller gap to capacity on a channel with ISI than on the ideal channel (that is  $\gamma_{\text{isi}} > 1$ ). Explain why this conclusion would be wrong.

## REFERENCES

1. M. E. Austin, *Decision-Feedback Equalization for Digital Communication Over Dispersive Channels*, M.I.T. Lincoln Laboratory, Lexington, Mass (August 1967).
2. C. A. Belfiore and J. H. Park, "Decision Feedback Equalization," *Proceedings of the IEEE* 67(8)(Aug. 1979).
3. M. Tomlinson, "New Automatic Equalizer Employing Modulo Arithmetic," *Electronic Letters* 7(March 1971).
4. H. Harashima and H. Miyakawa, "A Method of Code Conversion for a Digital Communication Channel with Intersymbol Interference," *Transactions. Institute Electronic Communication Engineering (Japan)* 52-A(June 1969).
5. H. Harashima and H. Miyakawa, "Matched-Transmission Technique for Channels with Intersymbol Interference," *IEEE Trans. on Communications* COM-20 p. 774 (Aug. 1972).
6. D. G. Messerschmitt, "Generalized Partial Response for Equalized Channels with Rational Spectra," *IEEE Trans. on Communications* COM-23(11) p. 1251 (Nov. 1975).
7. A. Lender, "Correlative Level Coding for Binary-Data Transmission," *IEEE Spectrum* 3 p. 104 (Feb. 1966).
8. G. D. Forney, Jr, "Coset Codes - Part I: Introduction and Geometrical Classification," *IEEE Trans. Information Theory* IT-34 p. 1123 (1988).
9. J. M. Cioffi, G. P. Dudevoir, M. Vedat Eyuboglu, and G. D. Forney, Jr, *MMSE Decision-Feedback Equalizers and Coding. Part I: General Results*. to appear.
10. R. W. Lucky, "Signal Filtering with the Transversal Equalizer," *Proc. Seventh Annual Allerton Conference on Circuits and System Theory*, p. 792 (Oct. 1969).
11. R. D. Gitlin and S. B. Weinstein, "Fractionally-Spaced Equalization: An Improved Digital Transversal Equalizer," *BSTJ* 60(2)(Feb. 1981).
12. S. U. H. Qureshi and G. D. Forney, Jr., "Performance Properties of a T/2 Equalizer," *NTC '77 Proceedings*, 0.

13. G. Ungerboeck, "Fractional Tap-Spacing and Consequences for Clock Recovery in Data Modems," *IEEE Trans. on Communications*, (Aug. 1976).
14. J. E. Mazo, "Optimum Timing Phase for an Infinite Equalizer," *BSTJ* 54(1)(Jan. 1975).
15. S.U.H.Qureshi, "Adaptive Equalization," pp. 640 in *Advanced Digital Communications Systems and Signal Processing Techniques*, ed. K. Feher, Prentice-Hall, Englewood Cliffs, N.J. (1987).
16. R. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, Inc., New York (1968).
17. J. R. Barry, E. A. Lee, and D. G. Messerschmitt, "Capacity Penalty Due to Ideal Tail-Canceling Equalization," *IEEE Trans. on Information Theory*, (to appear).
18. J. R. Barry, E. A. Lee, and D. G. Messerschmitt, "Capacity Penalty Due to Ideal Zero-Forcing Decision-Feedback Equalization," *Proceedings Int. Conf. Communications*, (June 1993).
19. R. Price, "Nonlinearly Feedback-Equalized PAM vs. Capacity for Noisy Filter Channels," *Proc. 1972 IEEE International Conf. Communications*, pp. 22-12 (June 1972).
20. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York (1965).
21. J. G. Proakis, *Digital Communications, Second Edition*, McGraw-Hill Book Co., New York (1989).
22. J. G. Proakis, "Advances in Equalization for Intersymbol Interference," *Advances in Communication Systems* 4(1975).
23. G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. on Information Theory* IT-18 pp. 363-378 (May 1972).
24. G. D. Forney, Jr., "The Viterbi Algorithm," *Proceedings of the IEEE* 61(3)(March 1973).
25. M. V. Eyuboglu and G. D. Forney, Jr., "Trellis Precoding: Combined Coding, Precoding, and Shaping for Intersymbol Interference Channels," *IEEE Trans. Information Theory*, (March 1992).
26. J. G. Proakis, "Adaptive Nonlinear Filtering Techniques for Data Transmission," *IEEE Symposium on Adaptive Processes, Decision, and Control*, p. XV.2.1 (1970).
27. A. Gersho and T. L. Lim, "Adaptive Cancellation of Intersymbol Interference for Data Transmission," *BSTJ* 70 pp. 1997-2021 (Nov. 1981).
28. D. D. Falconer and F. R. Magee, Jr., "Adaptive Channel Memory Truncation for maximum Likelihood Sequence Estimation," *Bell Sys. Tech. J.* 52 p. 1541 (Nov. 1973).
29. W. U. Lee and F. S. Hill, "A Maximum-Likelihood Sequence Estimator with Decision Feedback Equalization," *IEEE Trans. on Communications* COM-25(9) pp. 971-979 (Sep. 1977).
30. K. Wesolowski, "An Efficient DFE & ML Suboptimum Receiver for Data Transmission Over Dispersive Channels Using Two-Dimensional Signal Constellations," *IEEE Trans. on Communications* COM-35(3) pp. 336-339 (March 1987).
31. D. L. Duttweiler, J. E. Mazo, and D. G. Messerschmitt, "Error Propagation in Decision-Feedback Equalizer," *IEEE Trans. on Information Theory* IT-20 pp. 490-497 (Jul. 1974).

# 11

---

## ADAPTIVE EQUALIZATION

---

In Chapter 8 we derived a set of receiver structures that counter intersymbol interference under the assumptions of a known channel and unconstrained implementation complexity. The resulting structures are impractical for most applications in the exact form we derived them for several reasons. First, assumption of a known received pulse shape is unrealistic, particularly for channels such as the digital subscriber loop (with bridged taps), radio channel (with selective fading), and voiceband data channel, where there are significant variations in the channel affecting the reception. Thus, the received pulse shape is not actually known in advance for these channels, and is sometimes varying during actual transmission. Second, the receiver structures we derived usually have an infinite number of coefficients, and cannot be realized. Third, our optimizations did not take into account significant impairments such as timing jitter and timing offset, which must be considered in the design of receive filtering.

Timing offset and jitter will be considered in Chapter 17. In this chapter we will address the problem of estimating the actual channel isolated pulse response, and automatically adjusting an equalizer to equalize this channel. This is known as *adaptive equalization*, and was first proposed and analyzed by R.W. Lucky in 1965 [1,2,3], building on earlier work in adaptive filtering by B. Widrow and M.E. Hoff, Jr. in 1960 [4]. Our general approach will be to define practical filter structures similar to those found to be optimal in Chapter 10, and then arrange to adapt the parameters of those structures to the actual channel characteristics.



The simplest form of adaptive equalizer is shown in Figure 11-1 in block diagram form. The received signal is applied to a receive filter. Since the channel is not assumed to be known, the receive filter is usually *not* a matched filter, although it may be a compromise approximation. Rather, it is more likely to be a lowpass filter which simply rejects all out-of-band noise. The output of the receive filter is sampled, usually at the symbol rate or twice the symbol rate, and applied to an *adaptive equalizer*. The adaptive equalizer may be realized as a *finite transversal filter*, which is a version of the transversal filter encountered in Section 10.4, with a finite number of taps or coefficients. The object is to *adapt* the coefficients to minimize the noise and intersymbol interference at the output, which is applied to a slicer to make the decisions on the data symbols. The adaptation of the equalizer is driven by an *error signal*, which indicates to the equalizer the direction that the coefficients must be moved to more accurately represent the data symbols at the slicer input.

In the steady state, the adaptation of the equalizer is *decision directed*. This means that the receiver decisions are used to generate the error signal. In the absence of intersymbol interference and noise, the slicer input would precisely equal the transmitted data symbols, and the slicer output would equal the slicer input. Thus, there would be no error, and the error signal at the adaptive equalizer would be zero. This would tell the equalizer that no adjustment of coefficients is necessary. If there were noise alone at the slicer input, but no intersymbol interference, the error signal would be non-zero, but would average to zero resulting in no net change in the coefficients. But when there is intersymbol interference, the resulting error signal can be used to adjust the coefficients so as to reduce that intersymbol interference. A more detailed block diagram for the passband PAM case was shown in Figure 6-23.

The adaptive equalizer thus uses the regenerative effect (Chapter 1) to advantage: since the slicer regenerates a noise- and intersymbol interference-free representation of the transmitted data symbols, a comparison of these symbols with the slicer input can be used to adjust the equalizer. Of course, the slicer makes occasional errors, but due to the long averaging time of the equalizer coefficient adjustment algorithm these errors have no significant effect.

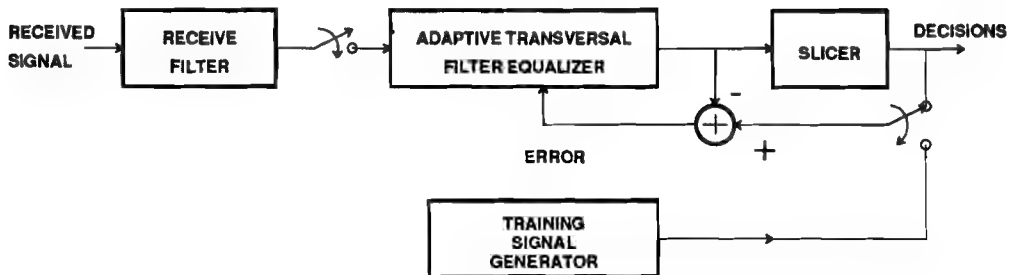


Figure 11-1. Block diagram of an adaptive equalizer.

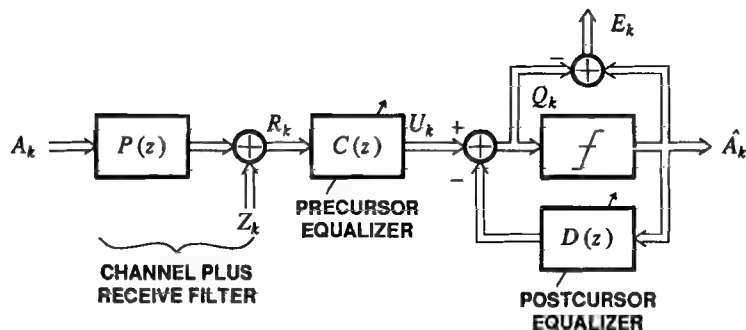
Decision-directed equalizer adjustment is effective in tracking slow variations in the channel response. It is often, however, not effective during initial acquisition since the intersymbol interference can be so bad as to cause a very high error rate initially. For this reason the initial acquisition of the equalizer is often accomplished by using a *training signal*. In this mode of operation, the transmitter generates a data symbol sequence known to the receiver. The receiver therefore substitutes this known training signal in place of the slicer output, as shown in Figure 11-1. Once an agreed period of time has elapsed, the slicer output is substituted and actual data transmission begins.

## 11.1. CONSTRAINED-COMPLEXITY EQUALIZERS

Before we can adapt an equalizer, we must specify the *structure* of a suitable filter that has *finite degrees of freedom* or *constrained complexity*, and only then can we design an algorithm for adapting the parameters of that structure. The constrained complexity transversal filter defined in Section 10.4 is suitable for that purpose. Using this structure, in this section we re-solve the optimization of the filter coefficients under the mean-square error (MSE) criterion for the linear equalizer (LE) case. This requires the solution of a set of linear equations; we discuss a specific method to obtain this solution called the *mean-square error gradient (MSEG) algorithm*. This method is of interest because it leads directly to an *adaptation algorithm* in Section 11.2.

### 11.1.1. Equalizer Structure

A block diagram of a complete adaptive decision-feedback equalizer system is shown in Figure 11-2. The coefficients of the equivalent discrete-time channel,  $P(z)$ , will in general be complex-valued, and the additive noise  $Z_k$  will also be complex-valued. The channel response and receive filter are reflected in the equivalent



**Figure 11-2.** An adaptive decision-feedback equalizer. A linear equalizer results when  $D(z) = 0$ .

discrete-time response  $P(z)$ . This channel model assumes that the demodulation has been performed prior to equalization, as was derived in Chapter 10, which is known as a *baseband adaptive equalizer*. In practice a *passband equalizer* is usually used, as discussed in Section 11.5, but we discuss the baseband case first since it is easier to understand and also models the important baseband channel case. We have also assumed symbol-rate sampling in the equalizer; the fractionally spaced case (Section 10.3) will be deferred to Section 11.4.

The DFE consists of a precursor equalizer and a postcursor equalizer; the postcursor equalizer is absent in the LE ( $D(z) = 0$ ). The error signal between slicer input and output,

$$E_k = \hat{A}_k - Q_k, \quad (11.1)$$

will be used to adapt the precursor and postcursor equalizers in the decision-directed mode; during the training mode the actual transmitted data symbol  $A_k$  is substituted for the decision  $\hat{A}_k$ . In the analysis of the equalizer, we assume that there are no decision errors, and thus use  $A_k$  in place of  $\hat{A}_k$ . This assumption is justified by experience, which confirms that as long as the error rate is below ten percent or so there is no appreciable effect on equalizer operation.

### Linear Equalizer (LE) Structure

For the LE in Chapter 10 the precursor equalizer is non-causal, even with unconstrained complexity, and thus it is natural to assume that the finite transversal filter has taps symmetrically spaced around the zero-delay tap. If we assume the total number of coefficients  $N$  is odd, and let  $L = (N-1)/2$ , then this equalizer would be of the form

$$C(z) = \sum_{m=-L}^L c_m z^{-m} \quad (11.2)$$

where  $c_m$ ,  $-L \leq m \leq L$  are the  $N$  coefficients of the precursor equalizer. This filter is not causal, but as discussed in Section 10.4 can always be made causal at the expense of an additional delay through the equalizer.

### Decision-Feedback Equalizer (DFE) Structure

For the DFE, in Chapter 10 the unconstrained complexity precursor equalizer was anti-causal in order to cancel precursor intersymbol interference, so we can assume an anti-causal  $N$  coefficient equalizer of the same form,

$$C(z) = \sum_{m=-(N-1)}^0 c_m z^{-m}. \quad (11.3)$$

The postcursor equalizer must be a strictly causal filter with  $M$  taps, viz.

$$D(z) = \sum_{m=1}^M d_m z^{-m}. \quad (11.4)$$

The number of feedforward coefficients  $N$  is allowed to be different from the number of postcursor equalizer coefficients  $M$ .

### 11.1.2. Minimum MSE Solution

We now return to the same problem solved in Chapter 10, the known channel case, and re-derive the optimal equalizer for constrained complexity, using the *minimum mean-square error (MSE)* criterion. We also make the assumption that all signals are wide-sense stationary. The solution will lead directly to a method of adapting the equalizer in Section 11.2. We specialize to the LE case here, and defer the DFE until Section 11.4.

In Chapter 10, the ZF criterion forced the intersymbol interference to zero, while the MSE criterion allowed intersymbol interference in order to reduce the noise variance at the slicer input. For the constrained complexity transversal filter, intersymbol interference is inevitable since there are insufficient degrees of freedom to get completely rid of it. Thus the ZF criterion is a little different — it now forces the intersymbol interference to zero for only a finite set of precursors and postcursors. The MSE criterion is much the same as before — it minimizes the MSE at the slicer input. Since intersymbol interference is inevitable, the constrained complexity MSE criterion makes more sense and is usually the one used. Therefore, we will consider only that criterion and will leave the ZF criterion to the problems.

For the transversal filter it is appropriate to use vector and matrix notation. Define a vector of transversal filter coefficients

$$\mathbf{c}' = [c_{-L}, \dots, c_L] \quad (11.5)$$

where  $\mathbf{c}'$  denotes the transpose of vector  $\mathbf{c}$ ,  $L = (N-1)/2$ , and  $N$  is the number of equalizer taps, assumed odd. Also define a vector of past and future input samples to the equalizer,

$$\mathbf{r}_k' = [R_{k+L}, \dots, R_k, \dots, R_{k-L}]. \quad (11.6)$$

If the slicer output is assumed to equal the actual transmitted data symbols (no decision errors), then the error signal is

$$E_k = A_k - Q_k, \quad Q_k = \mathbf{c}' \mathbf{r}_k. \quad (11.7)$$

In general all these quantities are complex-valued, although for the baseband channel they are all real-valued. Assume that all the random processes are wide-sense stationary with known statistics, and design the equalizer coefficient vector  $\mathbf{c}$  to minimize the MSE, defined as  $E[|E_k|^2]$ .

#### Exercise 11-1.

Explicitly evaluate the mean-square error and show that it is equal to

$$\begin{aligned} E[|E_k|^2] &= E[|A_k|^2] - 2\text{Re}\{\mathbf{c}'^* E[A_k \mathbf{r}_k^*]\} + \mathbf{c}'^* E[\mathbf{r}_k \mathbf{r}_k'] \mathbf{c} \\ &= E[|A_k|^2] - 2\text{Re}\{\mathbf{c}'^* \alpha\} + \mathbf{c}'^* \Phi \mathbf{c}, \end{aligned} \quad (11.8)$$

where  $\alpha$  and  $\Phi$  are defined as

$$\alpha = E[A_k \mathbf{r}_k^*] \quad (11.9)$$

$$\Phi = E[\mathbf{r}_k^* \mathbf{r}_k'] = \begin{bmatrix} \phi_0 & \phi_{-1} & \cdots & \phi_{-(N-1)} \\ \phi_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \phi_{N-1} & \cdots & \cdots & \phi_0 \end{bmatrix}, \quad \phi_j = E[R_{k+j} R_k^*]. \quad (11.10)$$

□

**Exercise 11-2.**

Assume the transmitted data symbols  $A_k$  are zero-mean, uncorrelated with each other, and have variance  $\sigma_a^2$ . Show that the autocorrelation function of the equalizer input is, for the channel model of Figure 11-2,

$$\phi_j = \sigma_a^2 \sum_k p_{k+j} p_k^* + 2N_0 \rho_f(j) \quad (11.11)$$

where the impulse response of the receive filter is  $f(t)$  and  $\rho_f(k)$  is the autocorrelation of the receive filter impulse response. □

The MSE is not a function of time due to the wide-sense stationarity assumption. The autocorrelation function  $\phi_j$  is just a simplified notation for  $R_R(j)$ , and  $\Phi$  is an *auto-correlation matrix* for the sampled data signal.

**Exercise 11-3.**

Show that  $\Phi$  has the following important properties:

- (a)  $\Phi$  is a *Hermitian matrix*, i.e.

$$\Phi^{*'} = \Phi. \quad (11.12)$$

- (b)  $\Phi$  is a *Toeplitz matrix*, i.e. the  $i, j$  element is a function of  $i-j$ . [5,6].

- (c)  $\Phi$  is a positive semidefinite matrix, i.e. the *Hermitian form*  $\mathbf{x}^{*'} \Phi \mathbf{x}$  is real-valued for any vector  $\mathbf{x}$  and is also non-negative,

$$\mathbf{x}^{*'} \Phi \mathbf{x} \geq 0. \quad (11.13)$$

□

In most but not all applications it can be assumed that this autocorrelation matrix is positive definite, and hence nonsingular. Instances where it is singular will be discussed in Section 11.4, but for the time being assume it is nonsingular.

Our goal is to find the vector  $\mathbf{c}$  that minimizes (11.8). There are several ways to accomplish this; in the spirit of illustrating useful techniques we will demonstrate two approaches. The first approach is to express the MSE in a form for which the minimum MSE solution is obvious.

**Example 11-1.**

As an aid to intuition, it is often helpful to specialize to the degenerate case of a single real-valued coefficient  $c$ . This simplifies the equations dramatically, and yet reveals many of the interesting properties. If the input signals  $A_k$  and  $R_k$  are real-valued, then

(11.8) becomes

$$E[E_k^2] = E[A_k^2] - 2\alpha c + \phi_0 c^2 = E[A_k^2] - \frac{\alpha^2}{\phi_0} + \phi_0 \left(c - \frac{\alpha}{\phi_0}\right)^2 \quad (11.14)$$

where  $\alpha = E[A_k R_k]$  and  $\phi_0 = E[R_k^2]$ . Since the square term is non-negative, the MSE is minimized if it is zero, and the optimal coefficient is  $c_{\text{opt}} = \alpha/\phi_0$ .  $\square$

#### Exercise 11-4.

Verify by multiplying it out that

$$E[|E_k|^2] = E[|A_k|^2] - \alpha^{*'} \Phi^{-1} \alpha + (\Phi^{-1} \alpha - c)^{*'} \Phi (\Phi^{-1} \alpha - c). \quad (11.15)$$

is equivalent to (11.8).  $\square$

Since  $\Phi$  is positive semidefinite, the last term in (11.15) is non-negative (from (11.13)) and is minimized by the choice

$$c_{\text{opt}} = \Phi^{-1} \alpha. \quad (11.16)$$

Since only the last term in (11.15) depends on  $c$ , (11.16) also minimizes the mean-square error, which has a resultant minimum value

$$\xi_{\min} = E[|A_k|^2] - \alpha^{*'} \Phi^{-1} \alpha = E[|A_k|^2] - \alpha^{*'} c_{\text{opt}}. \quad (11.17)$$

Finding  $c_{\text{opt}}$  from (11.16) requires the solution of a system of linear equations. Under our nonsingular  $\Phi$  assumption, the solution of these equations is unique.

We now have a useful formula for the MSE, (11.15), which we can write in the form

$$E[|E_k|^2] = \xi_{\min} + (c - c_{\text{opt}})^{*'} \Phi (c - c_{\text{opt}}). \quad (11.18)$$

This consists of a term independent of the coefficient vector  $c$ , and a second term that is a Hermitian form in a vector  $(c - c_{\text{opt}})$  with matrix  $\Phi$ .

Another way to find the optimal solution is to take the gradient of the MSE with respect to the coefficient vector, and set that gradient to zero to find  $c_{\text{opt}}$ . For the baseband channel case, where everything is real-valued, this is straightforward. In the complex-valued case we have to be more careful because derivatives of some innocuous-looking complex-valued functions do not exist (for example, the derivative of  $z^*$ , the conjugate of a complex variable  $z$ , does not exist anywhere!). Fortunately, with the MSE we are dealing with a real-valued function of a complex vector  $c$ , which makes life simpler because we can consider the MSE to be a real-valued function of two real-valued vectors (the real and imaginary parts of  $c$ ).

If we define the real and imaginary parts of the complex quantities in (11.8),

$$c = c_R + j c_I, \quad \alpha = \alpha_R + j \alpha_I, \quad \Phi = \Phi_R + j \Phi_I, \quad (11.19)$$

and consider the real-valued function  $E[|E_k|^2]$  to be a function of two real-valued vectors  $c_R$  and  $c_I$ .

**Exercise 11-5.**

- (a) Show that as a consequence of the Hermitian property,

$$\Phi_R = \Phi_R^*, \quad \Phi_I = -\Phi_I^*. \quad (11.20)$$

- (b) Show that

$$\nabla_{\mathbf{c}_R} \mathbf{c}^* \Phi \mathbf{c} = 2\Phi_R \mathbf{c}_R - 2\Phi_I \mathbf{c}_I, \quad \nabla_{\mathbf{c}_I} \mathbf{c}^* \Phi \mathbf{c} = 2\Phi_R \mathbf{c}_I + 2\Phi_I \mathbf{c}_R, \quad (11.21)$$

$$\nabla_{\mathbf{c}_R} \text{Re}\{\mathbf{c}^* \alpha\} = \alpha_R, \quad \nabla_{\mathbf{c}_I} \text{Re}\{\mathbf{c}^* \alpha\} = \alpha_I, \quad (11.22)$$

where  $\nabla_{\mathbf{x}}$  is the gradient with respect to vector  $\mathbf{x}$ .

- (c) Show that if we define a gradient of a real-valued function with respect to a complex vector
- $\mathbf{c}$
- as

$$\nabla_{\mathbf{c}} = \nabla_{\mathbf{c}_R} + j \nabla_{\mathbf{c}_I} \quad (11.23)$$

then

$$\nabla_{\mathbf{c}} \mathbf{c}^* \Phi \mathbf{c} = 2\Phi \mathbf{c}, \quad \nabla_{\mathbf{c}} \text{Re}\{\mathbf{c}^* \alpha\} = \alpha. \quad (11.24)$$

□

Given the results of the exercise, we can find  $\mathbf{c}_{\text{opt}}$  by taking the gradients of  $E[|E_k|^2]$  from (11.8) with respect to  $\mathbf{c}_R$  and  $\mathbf{c}_I$  and setting them to zero to find the real and imaginary parts of  $\mathbf{c}_{\text{opt}}$ . In view of our definition of a gradient with respect to  $\mathbf{c}$ , this is equivalent to

$$\nabla_{\mathbf{c}} E[|E_k|^2] = 2\Phi \mathbf{c} - 2\alpha = 0, \quad (11.25)$$

which yields the same solution as (11.16).

**Orthogonality Principle**

If we calculate the crosscorrelation between the slicer error signal and the input signal to the equalizer, assuming the equalizer coefficient vector is optimal,

$$\begin{aligned} E[E_k \mathbf{r}_k^*] &= E[(A_k - \mathbf{c}_{\text{opt}}^* \mathbf{r}_k) \mathbf{r}_k^*] = E[A_k \mathbf{r}_k^* - \mathbf{r}_k^* \mathbf{r}_k^* \mathbf{c}_{\text{opt}}] \\ &= \alpha - \Phi \mathbf{c}_{\text{opt}} = 0. \end{aligned} \quad (11.26)$$

This result is known as the *orthogonality principle*. This principle is the first inkling of a possible approach to adapting the equalizer. In particular, it tells us a way in which the filter can tell if the coefficients are optimal; namely, the different delays of the sampled data signal at the input should be orthogonal to the slicer error. If this condition is not met, then the orthogonality principle doesn't necessarily tell us which direction to move the coefficients to bring them closer to the optimum, but we will find such a method in the next section.

**11.1.3. The MSE Gradient Algorithm**

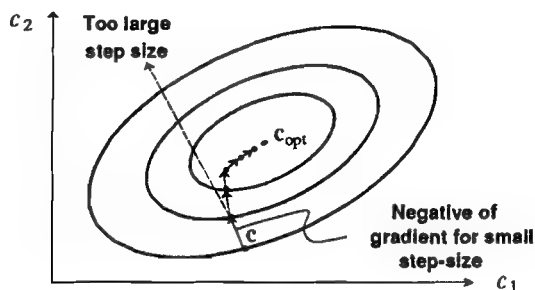
The previous results indicate that a system of linear equations must be solved in order to find the optimal MSE coefficient vector. Any number of numerical techniques could be used to solve these equations, but we now focus on the MSE gradient

(MSEG) algorithm. This method is of interest because it leads directly to an adaptive algorithm for equalizer adjustment, the *stochastic gradient (SG) algorithm*, in Section 11.2. Further, understanding of the convergence properties of the MSEG algorithm is a prerequisite to the understanding of the SG algorithm.

The MSEG algorithm defines a sequence of coefficient vectors that is guaranteed to converge to  $\mathbf{c}_{\text{opt}}$ , assuming that a unique optimum exists. As a starting point to the derivation,  $R_k$  is again assumed to be wide-sense stationary with nonsingular autocorrelation matrix (11.10). The output MSE given by (11.8) is a quadratic form in the coefficient vector and therefore has a unique global minimum. The MSE can be viewed as a surface in  $(N+1)$ -dimensional space (since it is a function of  $N$  coefficients). The quadratic nature of this surface makes it simple to adjust the weights iteratively to minimize the MSE by descending along the MSE surface. This is the MSE gradient algorithm.

Since the algorithm is iterative in nature, a notation for the coefficient vector that reflects this is needed. Thus, call the  $j$ -th iteration of the coefficient vector  $\mathbf{c}_j$ . Given the present coefficient vector  $\mathbf{c}_j$ , by subtracting off a term proportional to the error gradient,  $\nabla_{\mathbf{c}} E[|E_k|^2]$ , the resultant tap vector should be closer to  $\mathbf{c}_{\text{opt}}$ . This is because the gradient of the error is a vector in the direction of maximum increase of the error. Moving a short distance in the opposite (negative) direction of the gradient should therefore reduce the error. On the other hand, moving too far in that direction might actually overshoot the minimum, and result in instability.

The MSEG algorithm is illustrated in Figure 11-3 for the order two case ( $N = 2$ ). Because of the quadratic nature of the MSE, the contours of constant mean-square error are elliptical. The negative of the gradient points in the direction of maximum decrease of the mean-square error. When the step size is small, the mean-square error is reduced at each step of the algorithm, and approaches the minimum of (11.17) asymptotically. When the step size is too large, as shown in Figure 11-3, the mean-square error can actually increase, and the algorithm becomes unstable.



**Figure 11-3.** The elliptical contours of equal MSE, and an illustration of the MSEG algorithm.



The MSEG algorithm is explicitly

$$\mathbf{c}_{j+1} = \mathbf{c}_j - \frac{\beta}{2} \nabla_{\mathbf{c}_j} E[|E_k|^2], \quad (11.27)$$

where  $\beta$  is a small adaptation constant or step size that controls the size of the change in  $\mathbf{c}_j$  at each update. The division by two is included to avoid a factor of two in the subsequent adaptation algorithm. From (11.25), this algorithm becomes

$$\mathbf{c}_{j+1} = \mathbf{c}_j + \beta(\alpha - \Phi \mathbf{c}_j) = (\mathbf{I} - \beta\Phi)\mathbf{c}_j + \beta\alpha \quad (11.28)$$

where  $\mathbf{I}$  is the identity matrix. Hopefully, if this algorithm is simply iterated from some arbitrary initial guess  $\mathbf{c}_0$  it will converge to  $\mathbf{c}_{\text{opt}}$  of (11.16).

### Example 11-2.

Continuing Example 11-1, in this case (11.28) becomes

$$c_{j+1} = (1 - \beta\phi_0)c_j + \beta\alpha \quad (11.29)$$

where  $c_j$  is the  $j$ -th iteration of a single real-valued coefficient  $c$ . It is simple to find a formula for  $c_j$  from (11.29), but even easier to subtract the optimal coefficient from Example 11-1 from both sides to obtain

$$(c_{j+1} - c_{\text{opt}}) = (1 - \beta\phi_0)(c_j - c_{\text{opt}}) = (1 - \beta\phi_0)^j (c_0 - c_{\text{opt}}), \quad (11.30)$$

which demonstrates that  $c_j \rightarrow c_{\text{opt}}$  as long as  $|1 - \beta\phi_0| < 1$ . Thus, the step size of the algorithm must be in the range  $0 < \beta < 2/\phi_0$  for there to be convergence. The convergence is exponential in the iteration index. The fastest convergence is for  $\beta = 1/\phi_0$ , in which case the MSEG algorithm converges in a single iteration!  $\square$

This simple example is readily extended to the general case. If  $\mathbf{c}_{\text{opt}}$  from (11.16) is subtracted from both sides of (11.28) and

$$\mathbf{q}_j = \mathbf{c}_j - \Phi^{-1}\alpha \quad (11.31)$$

is defined as the error between the actual and optimal coefficient vector, then

$$\mathbf{q}_{j+1} = (\mathbf{I} - \beta\Phi)\mathbf{q}_j = (\mathbf{I} - \beta\Phi)^{j+1}\mathbf{q}_0. \quad (11.32)$$

The question becomes whether this error converges to zero.

The behavior of (11.32) depends critically on the eigenvalues of the matrix  $\Phi$ , which we denote by  $\lambda_1, \dots, \lambda_n$ , and which are explored in the following exercise.

### Exercise 11-6.

Let  $\mathbf{A}$  be a Hermitian matrix. Then establish the following facts:

- The eigenvalues of  $\mathbf{A}$  are real-valued.
- Let  $\lambda_i$  and  $\lambda_j$  be two distinct (real-valued) eigenvalues of  $\mathbf{A}$ . Show that the associated eigenvectors  $\mathbf{v}_i$  and  $\mathbf{v}_j$  are orthogonal; that is,  $\mathbf{v}_i^H \mathbf{v}_j = 0$ .
- Assume for simplicity that the eigenvalues of  $\mathbf{A}$  are all distinct, and that the eigenvectors are normalized to unit length ( $\mathbf{v}_i^H \mathbf{v}_i = 1$ ). Define the matrix

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N] \quad (11.33)$$

called the *modal matrix*. Show that this modal matrix is *unitary*, i.e.

$$\mathbf{V}^{-1} = \mathbf{V}^{*'} \quad (11.34)$$

(d) Show that

$$\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^{*'} \quad (11.35)$$

where  $\mathbf{\Lambda}$  is a diagonal matrix of eigenvalues,

$$\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N) \quad (11.36)$$

(e) Show that if  $\mathbf{A}$  is positive definite (semi-definite) then its eigenvalues are all positive (non-negative).  $\square$

The results of the exercise generalize to the case where the eigenvalues are not distinct [7]. In particular, if an eigenvalue is repeated with multiplicity  $m$ , then there are  $m$  corresponding linearly independent eigenvectors. These can always be constructed (by the Gram-Schmidt procedure, for example) to be mutually orthogonal and orthogonal to all the other eigenvectors. Thus, the decompositions of (11.35) apply to a general Hermitian matrix. More precisely, for any Hermitian matrix  $\Phi$  there exists a unitary matrix  $\mathbf{V}$  such that the diagonalizing transformation of (11.35) holds.

#### Exercise 11-7.

These results can be applied to our autocorrelation matrix  $\Phi$ , which is Hermitian from Exercise 11-3. Let  $\lambda_1 \cdots \lambda_N$  be the real-valued non-negative eigenvalues of  $\Phi$  and let  $\mathbf{v}_1 \cdots \mathbf{v}_N$  be a set of associated eigenvectors chosen to be mutually orthogonal. Establish that  $\Phi$  can be written in the form

$$\Phi = \sum_{i=1}^N \lambda_i \mathbf{v}_i \mathbf{v}_i^{*'} \quad (11.37)$$

This is known as a *spectral decomposition* of the matrix.  $\square$

#### Exercise 11-8.

Show that the eigenvectors of the matrix  $(\mathbf{I} - \beta\Phi)$  are the same as the eigenvectors of  $\Phi$ , and that the eigenvalues are  $(1 - \beta\lambda_i)$ ,  $1 \leq i \leq N$ , and thus establish the following decomposition, known as a *modal decomposition*,

$$(\mathbf{I} - \beta\Phi)^j = \sum_{i=1}^N (1 - \beta\lambda_i)^j \mathbf{v}_i \mathbf{v}_i^{*'} \quad (11.38)$$

The  $i$ -th term in (11.38) is known as the  *$i$ -th mode of the convergence*.  $\square$

From (11.38) and (11.32), the error vector  $\mathbf{q}_j$  obeys a trajectory that is the sum of  $N$  modes, the  $i$ -th of which is proportional to  $(1 - \beta\lambda_i)^j$ . The speed of convergence of each of these modes is governed by  $\beta$ . If  $\beta$  is made too large, then one or more of the  $(1 - \beta\lambda_i)$  terms will be larger than unity in magnitude, and the error vector in (11.38) will actually increase in size with time. This is quite consistent with the intuitive behavior exhibited in Figure 11-3 since the large  $\beta$  causes an overshoot of the minimum and actually increases the error.

This acceptable range of  $\beta$  can be investigated further if we order the eigenvalues from smallest to largest, denoting the smallest as  $\lambda_{\min}$  and the largest as  $\lambda_{\max}$ . Then the  $(1 - \beta\lambda_i)$  term that governs how large  $\beta$  can get is the one corresponding to the largest eigenvalue, and hence the condition for  $q_j$  decaying exponentially to zero is

$$0 < \beta < \frac{2}{\lambda_{\max}}. \quad (11.39)$$

This determines the largest value of  $\beta$ , but of more interest is the  $\beta$  corresponding to the fastest convergence of the MSEG algorithm. For a fixed  $\beta$ , the speed of convergence of the algorithm can be considered to be dominated by the slowest converging mode in (11.38). This slowest mode corresponds to the largest value of  $|1 - \beta\lambda_i|$ . The two extreme cases are plotted in Figure 11-4, where this term is calculated for  $\lambda_{\min}$  and  $\lambda_{\max}$ . The corresponding curves for the other eigenvalues lie in between these two curves. The value of  $\beta$  that results in the fastest convergence is the point labeled  $\beta_{\text{opt}}$  in the figure. Choice of any other value of  $\beta$  results in a slower convergence of the mode corresponding to either the maximum or minimum eigenvalue. This optimal value of  $\beta$  is easily shown to be

$$\beta_{\text{opt}} = \frac{2}{\lambda_{\min} + \lambda_{\max}}, \quad (11.40)$$

and for this choice of  $\beta$  the modes corresponding to both minimum and maximum eigenvalues converge at the same rate, namely proportional to

$$\left[ \frac{\lambda_{\max}/\lambda_{\min} - 1}{\lambda_{\max}/\lambda_{\min} + 1} \right]^j. \quad (11.41)$$

The quantity in the parenthesis is plotted in Figure 11-5 as a function of the parameter  $\lambda_{\max}/\lambda_{\min}$ . This parameter, the ratio of largest to smallest eigenvalue, is called the *eigenvalue spread*. The eigenvalue spread has a minimum value of unity, and can be arbitrarily large. The larger the eigenvalue spread of the autocorrelation matrix, the slower the convergence of the MSEG algorithm. As seen in Figure 11-5, the convergence becomes arbitrarily slow as the eigenvalue spread approaches infinity since the quantity in parentheses in (11.41) approaches unity.

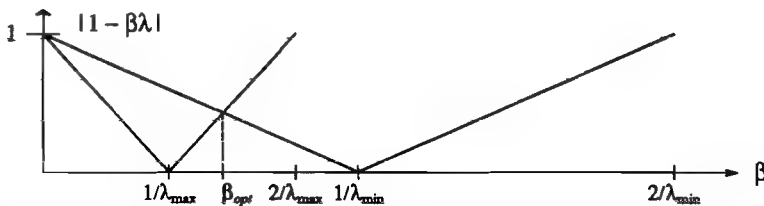


Figure 11-4. Choice of step size for fastest convergence of the MSEG algorithm.

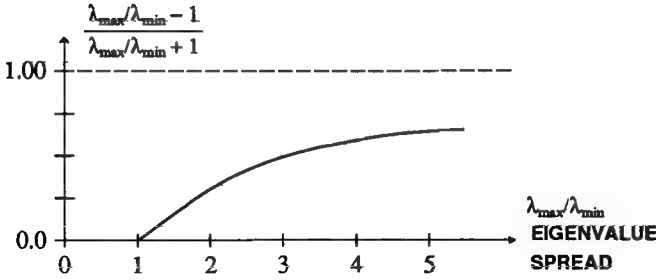


Figure 11-5. Relation of fastest convergence rate to eigenvalue spread.

The important role of the eigenvalues can be further quantified based on the following exercise.

**Exercise 11-9.**

Show that

$$\begin{aligned} E[|E_k|^2] - E[|E_k|^2]_{\min} &= (\mathbf{c}_k - \mathbf{c}_{\text{opt}})^* \Phi (\mathbf{c}_k - \mathbf{c}_{\text{opt}}) \\ &= \sum_{i=1}^n \lambda_i |(\mathbf{c}_k - \mathbf{c}_{\text{opt}})^* \mathbf{v}_i|^2, \end{aligned} \quad (11.42)$$

using expansion (11.38).  $\square$

Equation (11.42) decomposes the MSE into its components in the direction of each of the eigenvectors, and shows that the component in that direction is proportional to the corresponding eigenvalue. The MSE therefore increases most rapidly in the direction of the eigenvector corresponding to  $\lambda_{\max}$  and most slowly in the direction corresponding to  $\lambda_{\min}$ . The largest acceptable step size  $\beta$  is determined by the maximum eigenvalue, since the gradient will be the largest in the direction of the eigenvector corresponding to the largest eigenvalue, and the correction of the MSEG algorithm will thus be largest in that direction. As  $\beta$  gets larger, that is the direction in which the increment in the algorithm will first be so large as to actually increase the MSE.

The  $\beta_{\text{opt}}$  of (11.40) is optimized to speed the convergence of the coefficient vector. An alternative approach is to choose  $\beta$  to maximize the rate of convergence of the MSE. From (11.42), the MSE at time  $k = 1$  can be expressed in terms of the coefficient error  $\mathbf{q}_1$  as

$$E[|E_1|^2] - E[|E_1|^2]_{\min} = \sum_{i=1}^n \lambda_i |\mathbf{q}_1^* \mathbf{v}_i|^2, \quad (11.43)$$

which can then be expressed in terms of  $\mathbf{q}_0$  using (11.38) as

$$E[|E_1|^2] - E[|E_1|^2]_{\min} = \sum_{i=1}^n \lambda_i (1 - \beta \lambda_i)^2 |\mathbf{q}_0^* \mathbf{v}_i|^2. \quad (11.44)$$

To get the maximum reduction in MSE from step  $k = 0$  to step  $k = 1$ , we would minimize (11.44) with respect to  $\beta$ . Unfortunately, this requires some assumption

about the initial coefficient error  $\mathbf{q}_0$ . However, the largest contributor to the MSE is likely to be the term in the sum corresponding to the largest eigenvalue,  $\lambda_{\max}$ . A reasonable strategy is therefore to choose  $\beta$  to immediately force that term to zero. This is achieved by choosing  $\beta = 1/\lambda_{\max}$ . The conclusion is that (11.40) is not necessarily the best choice for step size, if the criterion is to most quickly reduce the MSE as opposed to reduce the norm of the coefficient error vector. The step size that speeds the convergence of the MSE depends on the initial coefficient error vector, but  $\beta = 1/\lambda_{\max}$  is a reasonable choice.

The convergence of the MSEG algorithm can be interpreted graphically by plotting the contours of equal mean-square error as in Figure 11-6. Equation (11.42) illustrates that the contours of equal mean-square error are elliptical in shape, with the principal axes in the direction of the eigenvectors. The eccentricities of the ellipses are directly related to the relative sizes of the eigenvalues. This is illustrated in Figure 11-6 for the  $N = 2$  case. It is assumed that  $\lambda_2 > \lambda_1$ , in which case the mean-square error increases more rapidly in the direction of  $\mathbf{v}_2$ . The direction of the two orthogonal eigenvectors is shown. The major axis of the ellipse is in the direction of  $\mathbf{v}_1$ , and the minor axis in the direction of  $\mathbf{v}_2$ .

The case where the eigenvalue spread is small (eigenvalues approximately equal) is shown in Figure 11-6a; the ellipse is close to being a circle. A larger eigenvalue spread is shown in Figure 11-6b; there the ellipse is more eccentric.

For a small eigenvalue spread, the gradient correction is always nearly in the direction of the minimum mean-square error, and the length of the gradient vector is always approximately the same. For a larger eigenvalue spread, the direction of the negative gradient can be quite different from the direction of the minimum, although for small steps the mean-square error still gets smaller. Since each step does not go directly toward the minimum, the number of required steps will be increased for some starting conditions. More importantly, the length of the gradient vector will be much smaller in the direction of the major axis of the ellipse, since the MSE is not varying as rapidly in that direction. The step size is therefore governed by the largest eigenvalue, so that the steps do not overshoot in the direction of the corresponding eigenvector, which is the minor axis of the ellipse. A step size that maintains stability along the minor axis results in very small increments in the direction of the major axis.

An intuitive interpretation of Figure 11-4 also follows from Figure 11-6. Consider the case where the starting coefficient vector is on the minor axis of the ellipse, so that convergence is in the direction of the eigenvector corresponding to the largest eigenvalue. If  $\beta$  is chosen to be smaller than  $1/\lambda_{\max}$ , then each step of the algorithm in this direction does not overshoot the minimum, and the MSE gets smaller. When  $\beta = 1/\lambda_{\max}$ , the algorithm converges to the minimum in one iteration. When  $\beta$  is greater than  $1/\lambda_{\max}$ , the algorithm overshoots on each iteration, but as long as  $\beta$  is smaller than  $2/\lambda_{\max}$  the MSE still decreases and the algorithm converges. It is advantageous from the point of view of maximizing the worst-case convergence rate to choose  $\beta = \beta_{\text{opt}}$ , a choice that results in the algorithm overshooting the minimum in the direction of the eigenvector corresponding to  $\lambda_{\max}$  in order that the algorithm

converge faster in the direction of the eigenvector corresponding to  $\lambda_{\min}$ .

Since the eigenvalue spread plays such an important role in the adaptation speed, it is instructive to relate it to the power spectral density of the wide-sense stationary random process  $R_k$ , the samples of the data waveform. It is a classical result of Toeplitz form theory [6] that the eigenvalues of (11.10) are bounded by

$$\min_{\omega} S(e^{j\omega}) < \lambda_i < \max_{\omega} S(e^{j\omega}), \quad (11.45)$$

where  $S(e^{j\omega})$  is the power spectral density of the reference random process defined as the Fourier transform of the autocorrelation function (the elements of the matrix),

$$S(e^{j\omega}) = \sum_{k=-\infty}^{\infty} \phi_k e^{-j\omega k}. \quad (11.46)$$

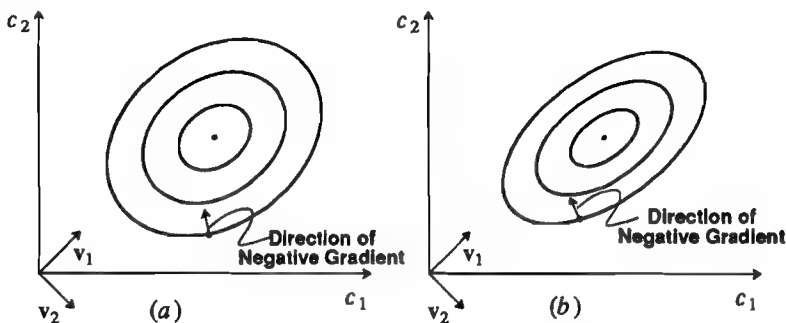
While the eigenvalues depend on the order of the matrix,  $n$ , as  $n \rightarrow \infty$

$$\lambda_{\max} \rightarrow \max_{\omega} S(e^{j\omega}), \quad \lambda_{\min} \rightarrow \min_{\omega} S(e^{j\omega}). \quad (11.47)$$

See [6] for more precise statements of these results. This interesting relationship between the eigenvalues and the power spectrum is explored in Problem 11-4.

It follows that the spectra that result in slow convergence of the MSEG algorithm are those for which the ratio of the maximum to the minimum of the spectrum is large, and spectra that are almost flat (have an eigenvalue spread near unity) result in fast convergence. The intuition behind this result is that a large eigenvalue spread is related to a large correlation among the input samples, which in turn slows convergence because of interactions between the convergence of the different coefficients of the transversal filter.

Since the modes of convergence of the MSEG algorithm are all of the form of  $\gamma^j$ , where  $\gamma$  is a positive real number less than unity and  $j$  is the iteration number, the error in decibels can be determined by taking the logarithm of the square (Problem



**Figure 11-6.** Effect of eigenvalue spread on convergence. a. Small eigenvalue spread. b. Larger eigenvalue spread.

11-8),

$$10 \log_{10}(\gamma^{2j}) = [10 \log_{10}(\gamma^2)] j, \quad (11.48)$$

and thus the error expressed in decibels decreases linearly with iteration number (the constant factors multiplying these exponentially decaying terms give a constant factor in decibels). The convergence of a MSEG algorithm is thus often expressed in units of *dB per iteration*, which is the number of decibels of decrease in the error power per iteration.

## 11.2. ADAPTIVE LINEAR EQUALIZER

On many practical channels, one cannot pretend to know the autocorrelation matrix  $\Phi$ , and hence the known-channel solution of Section 11.1 is not applicable.

### Example 11-3.

On the voiceband telephone channel (Section 5.5), there is significant variation in the amplitude and phase of the channel transfer function from one call to another. On a terrestrial microwave channel (Section 5.4), under normal conditions the channel is nearly ideal, but there can be conditions under which there is considerable variation in the transfer function due to selective fading.  $\square$

The technique is to modify the MSEG algorithm, by a simple trick, to allow adaptation. The resulting algorithm is known as the *stochastic gradient (SG)* algorithm. The approach taken in the SG algorithm is to substitute a time average for the ensemble average in the MSE solution. This adaptation algorithm is also sometimes called the LMS adaptive transversal filter. The term LMS stands for *least-mean square*, although the algorithm does not provide an exact solution to the problem of minimizing the mean-square error but rather only approximates the solution. This approximation is the price paid for not requiring that the channel be known or stationary.

If we don't know the channel, then we cannot calculate the expectation in (11.8). However, we can, if we choose, calculate the modulus-squared error without the expectation,

$$|E_k|^2 = |A_k|^2 - 2\text{Re}\{A_k \mathbf{c}^{*'} \mathbf{r}_k^*\} + \mathbf{c}^{*'} \mathbf{r}_k^* \mathbf{r}_k^* \mathbf{c}. \quad (11.49)$$

Using Exercise 11-5, and using the fact that  $\mathbf{r}_k^* \mathbf{r}_k^{'}$  is a Hermitian matrix, we can take the gradient of this expression,

$$\nabla_{\mathbf{c}} |E_k|^2 = -2\mathbf{r}_k^* (A_k - \mathbf{r}_k^* \mathbf{c}) = -2E_k \mathbf{r}_k^*. \quad (11.50)$$

Because we are dealing with well-behaved quadratic functions, and the gradient and expectation are linear operators, they can be interchanged. The expectation of the gradient in (11.50) is the same as the gradient of  $E[|E_k|^2]$ ,

$$E[\nabla_{\mathbf{c}} |E_k|^2] = \nabla_{\mathbf{c}} E[|E_k|^2]. \quad (11.51)$$

Therefore, (11.50) is an *unbiased estimator* of the gradient in the MSEG algorithm

derived in Section 11.1. The SG algorithm substitutes this "noisy" or "stochastic" gradient for the actual gradient in the algorithm of (11.28). What results is

$$\mathbf{c}_{k+1} = \mathbf{c}_k - \frac{\beta}{2} \nabla_{\mathbf{c}} [|\mathbf{E}_k|^2] \Big|_{\mathbf{c}=\mathbf{c}_k}, \quad (11.52)$$

or

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \beta \mathbf{E}_k \mathbf{r}_k^* = [\mathbf{I} - \beta \mathbf{r}_k^* \mathbf{r}_k'] \mathbf{c}_k + \beta \mathbf{A}_k \mathbf{r}_k^*. \quad (11.53)$$

There is another rather subtle but important difference between this SG algorithm and the algorithm of (11.28). In the former algorithm, the index  $j$  corresponded to the iteration number for the iterative algorithm for solving a system of linear equations. In (11.53) on the other hand, the iteration number  $k$  corresponds to the sample number (or time index) of the data waveform at the input to the equalizer. Thus each iteration corresponds to a new sample. The algorithm is in effect performing a time average in order to estimate the gradient.

It is not surprising to note the similarity between the SG algorithm of (11.53) and the MSEG algorithm of (11.28). The former substitutes the stochastic matrix  $\mathbf{r}_k^* \mathbf{r}_k'$  for  $\Phi$  and the stochastic vector  $\mathbf{A}_k \mathbf{r}_k^*$  for the vector  $\alpha$ . In each case the deterministic matrix or vector corresponds to the ensemble average of the stochastic matrix or vector for the stationary case.

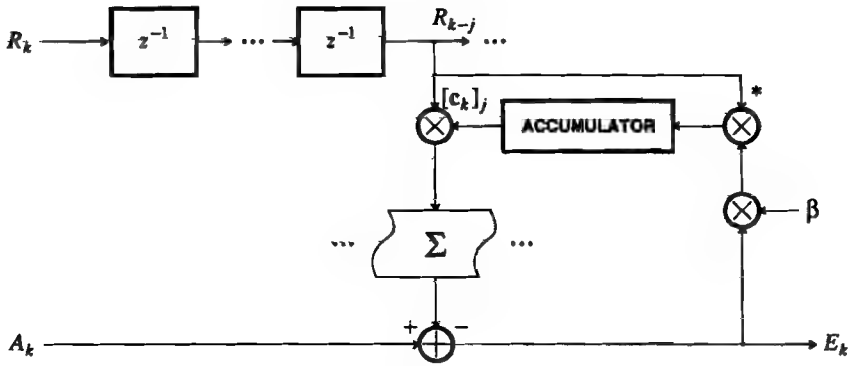
A realization of adaptation algorithm (11.53) is illustrated in Figure 11-7. What is shown is a single filter coefficient and how it contributes both to the transversal filter structure as well as how it is adapted. Denote the  $j$ -th coefficient  $c_j$  at time  $k$  by  $[c_k]_j$ . Then the adaptation algorithm of (11.53) can be rewritten for the  $j$ -th component as

$$[c_{k+1}]_j = [c_k]_j + \beta E_k R_{k-j}^*, \quad -L \leq j \leq L. \quad (11.54)$$

Specifically, the input sample to the equalizer  $R_{k-j}$  is taken from the output of the same unit delay as is used for multiplication by  $c_j$ , conjugated, and multiplied by the  $j$ -th coefficient at time  $k$ . The resultant value contributes to the summation which is subtracted from the data symbol  $A_k$  to obtain the error sample  $E_k$ . In accordance with this equation,  $[c_k]_j$  is obtained by cross-correlating (time averaging) the estimation error  $E_k$  with the delayed input  $R_{k-j}$ . This cross-correlation consists of taking the product of  $E_k$  with  $R_{k-j}^*$  and step size  $\beta$  and accumulating the result.

This algorithm is not surprising in light of the orthogonality principle of (11.26). In particular, when all the filter coefficients are optimal, the orthogonality principle says that the input to the accumulator in Figure 11-7 will average to zero. Under these conditions, the output of the accumulator will maintain the same average value (namely the optimal filter coefficient). What the orthogonality principle does not tell us directly is that when the coefficients are not optimal, the non-zero average of the accumulator input is of the correct sign so as to move each coefficient toward the optimum.





**Figure 11-7.** SG algorithm for one coefficient. Although signals are shown with single lines, in general all signals and coefficients are complex-valued.

#### Example 11-4.

Continuing Example 11-1 for a single real-valued coefficient  $c$ , denote this coefficient at time  $k$  as  $c_k$ , and then (11.54) becomes

$$c_{k+1} = c_k + \beta E_k R_k, \quad E_k = A_k - c_k R_k. \quad (11.55)$$

Now suppose that  $c_k = c_{\text{opt}} + \Delta$  for some  $\Delta > 0$ . Then the correction term in the algorithm is

$$E_k R_k = (A_k - c_{\text{opt}} R_k) R_k - \Delta R_k^2. \quad (11.56)$$

The first term in (11.56) has an average value of zero by the orthogonality principle, and the second term is *always* negative, decreasing the coefficient on average as desired. (In spite of the fact that the second term gives the correction a bias in the right direction, the correction is stochastic because of the first term, and hence will sometimes go in the wrong direction.)  $\square$

This example is easily generalized. The term in the product  $E_k R_{k-j}^*$  which depends on  $[c_k]_j$  is  $-[c_k]_j |R_{k-j}|^2$ . Consider for example the case where the real part of  $[c_{\text{opt}}]_j$  is positive. Since  $|R_{k-j}|^2$  is positive real-valued, if the real part of  $[c_k]_j$  is too large then the real part of this gradient term is more negative than it would be if the coefficient were optimal. This makes the average correction to the real part of  $[c_k]_j$  negative and on average the real part of  $[c_{k+1}]_j$  is smaller than  $[c_k]_j$ . The same logic applies to the imaginary part of  $[c_k]_j$ .

This explanation does not answer the important question as to the effect of the interaction between the adaptation of the different coefficients. This interaction occurs because all the coefficients affect the error  $E_k$ , which in turn affects the coefficient adaptation. We now address this question.

### 11.2.1. Convergence of the SG Algorithm

One difference between the SG algorithm of (11.53) and the MSEG algorithm of (11.28) is that in (11.28) the coefficient vector follows a deterministic and predictable trajectory, while the trajectory in (11.53) is random or stochastic. The cause of this random fluctuation is the use of the time average in place of ensemble average, or alternatively the use of the input samples in place of the ensemble averages.

A useful method for analyzing the convergence behavior of an adaptive filtering algorithm is to assume (often unrealistically) that the input samples can be modeled as a wide-sense stationary random process with known statistics; that is, return to the assumptions of Section 11.1. The coefficient vector can then be expected to converge in some sense to be determined to the MSE solution. The speed of this convergence, while not directly applicable to a case where the input statistics are actually changing with time, is a good indication of the convergence performance of the algorithm.

If the assumption is made that the input can be modeled as a wide-sense stationary random process, we would first like to find the average trajectory of the coefficient vector in (11.53). If the step size  $\beta$  is small, the coefficient vector will vary slowly, since the update at each sample time is proportional to  $\beta$ . The input random process will therefore vary rapidly relative to the coefficient vector. If we take the expectation of (11.53) with respect to the statistics of  $\mathbf{r}_k$ , it is therefore a good approximation to assume that the coefficient vector  $\mathbf{c}_k$  is a constant with respect to this expectation, and hence

$$\mathbf{c}_{k+1} = E[(\mathbf{I} - \beta \mathbf{r}_k^* \mathbf{r}_k) \mathbf{c}_k] + \beta E[A_k \mathbf{r}_k^*] \approx (\mathbf{I} - \beta \Phi) \mathbf{c}_k + \beta \alpha, \quad (11.57)$$

where the expectation is only with respect to the input process. Although the coefficient vector is varying slowly, it is still random, and hence we must still take the expectation of both sides with respect to the ensemble of coefficient vectors,

$$E[\mathbf{c}_{k+1}] \approx (\mathbf{I} - \beta \Phi) E[\mathbf{c}_k] + \beta \alpha. \quad (11.58)$$

This approximate average trajectory precisely obeys the earlier deterministic MSEG algorithm. It can be asserted without further analysis that within the accuracy of this approximation the average trajectory of the SG converges to the optimal coefficient vector under the same condition that guarantees convergence of the MSEG algorithm, and the nature of the convergence is identical to that discussed in Section 11.1.2.

This does not mean, however, that any particular coefficient vector trajectory itself converges to the optimum, but only that the average of all trajectories converges to the optimum. In fact, the coefficient vector does not converge to the optimum. Even after convergence of the coefficient vector in the mean-value sense, the difference equation (11.53) still has a stochastic driving term, and therefore the coefficient vector continues to fluctuate about the optimal coefficient vector randomly. The larger the value of the step size  $\beta$ , the larger this fluctuation. The size of this fluctuation will be considered in a moment. Keeping it reasonably small generally requires a much smaller step size than the value given by (11.40).

These considerations make it important to calculate some measure of the variation of the coefficient vector about this optimum. In analogy to (11.31), define an

error vector between the actual coefficient vector at time  $k$  and the optimal vector as

$$\mathbf{q}_k = \mathbf{c}_k - \mathbf{c}_{\text{opt}} \quad (11.59)$$

**Exercise 11-10.**

Substitute into the SG algorithm of (11.53) to show that the update equation for the error vector is

$$\mathbf{q}_{k+1} = \Gamma_k \mathbf{q}_k + \beta D_k \mathbf{r}_k^* \quad (11.60)$$

where  $\Gamma_k$  is a stochastic matrix

$$\Gamma_k = \mathbf{I} - \beta \mathbf{r}_k^* \mathbf{r}_k^T \quad (11.61)$$

and  $D_k$  is the error signal for a transversal filter with optimal coefficients,

$$D_k = A_k - \mathbf{r}_k^T \mathbf{c}_{\text{opt}} \quad (11.62)$$

This interesting relationship demonstrates again that the coefficient vector can never reach its optimum because this stochastic equation has a driving term which never goes to zero (except in the degenerate case where the optimal filter yields a zero error signal).  $\square$

One measure of how well the SG algorithm is working would be the Euclidean norm of the coefficient error vector,

$$\|\mathbf{q}_k\|^2 = \mathbf{q}_k^{*T} \mathbf{q}_k \quad (11.63)$$

This is the appropriate measure to use if the accuracy with which the algorithm approximates the optimal coefficients is the primary concern. If, on the other hand, we are interested in how well the adaptive filter does its job, as manifested by the size of the error signal (which after all is what causes incorrect decisions in the slicer), then we are interested in the size of  $E_k$  rather than the error vector.

Actually, as one might expect, these two measures are closely related. If the coefficient vector is fixed, then from (11.15),

$$E[|E_k|^2] = \xi_{\min} + \mathbf{q}^{*T} \Phi \mathbf{q} \quad (11.64)$$

where  $\xi_{\min}$  is the minimum MSE that would result from the use of the optimal coefficient vector  $\mathbf{c}_{\text{opt}}$  ( $\mathbf{q} = \mathbf{0}$ ). The second term in (11.64) we call the *excess MSE*, or that MSE over and above the minimum possible due to the non-optimal coefficient vector. If the step size  $\beta$  is very small, then within the time frame of significant variation in the input random process  $R_k$  we would expect very little change in  $\mathbf{q}$ . Thus, we can substitute the time-varying coefficient vector for the fixed vector in (11.64) to accurately find how the MSE varies with time,

$$E[|E_k|^2] \approx \xi_{\min} + \mathbf{q}_k^{*T} \Phi \mathbf{q}_k \quad (11.65)$$

In order to find the average MSE, we must average (11.65) over the ensemble of coefficient error vectors, or

$$E[|E_k|^2] \approx \xi_{\min} + E[\mathbf{q}_k^{*T} \Phi \mathbf{q}_k] \quad (11.66)$$

This equation establishes the link that we were looking for between the coefficient

error vector and the filter output MSE.

Rather than analyze the dynamics of (11.66) in general, we will specialize to an input process  $R_k$  with uncorrelated zero-mean samples. For this case, the autocorrelation matrix is diagonal,

$$\Phi = \phi_0 \mathbf{I}, \quad \phi_0 = E[|R_k|^2]. \quad (11.67)$$

The average MSE as a function of time from (11.66) becomes a simpler expression

$$E[|E_k|^2] = \xi_{\min} + \phi_0 E[\|\mathbf{q}_k\|^2] \quad (11.68)$$

that relates directly to the Euclidean norm of the coefficient error vector. Estimation of this norm is somewhat tedious, so we defer it to Appendix 11-A. Derived there is a difference equation in the error vector norm

$$E[\|\mathbf{q}_{k+1}\|^2] = \gamma E[\|\mathbf{q}_k\|^2] + \beta^2 N \phi_0 \xi_{\min}, \quad \gamma = 1 - 2\beta\phi_0 + \beta^2 N \phi_0^2. \quad (11.69)$$

There is only a single mode of adaptation due to the assumption of a white input spectrum (and therefore there is only one distinct eigenvalue). This relation demonstrates both the speed with which the error vector norm decreases with time as the filter is adapting and the asymptotic error vector norm, which is non-zero due to the continued stochastic driving term even after nominal convergence. This latter contribution to excess MSE is related to the minimum MSE for a fixed-coefficient filter because this error appears in the error signal driving adaptation even after nominal convergence of the coefficient vector.

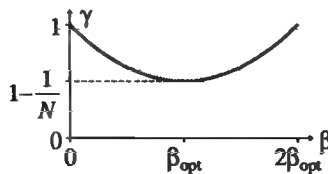
The condition for stability of the filter, in the sense that the error vector norm decreases with time, is that

$$|\gamma| < 1. \quad (11.70)$$

The quantity  $\gamma$  is plotted in Figure 11-8, where we see several interesting properties. Starting with zero, as we increase the step size  $\beta$  the speed of convergence increases, until a maximum speed is reached at

$$\beta_{\text{opt}} = \frac{1}{N\phi_0}. \quad (11.71)$$

Continuing to increase the step size slows convergence, until eventually we reach instability at twice the optimal step size. The condition for stability is



**Figure 11-8.** A plot of  $\gamma$  vs.  $\beta$  characterizing convergence of the MSE for a white input spectrum. A small  $N$  ( $N = 2$ ) is plotted to exaggerate the curve.

$$0 < \beta < \frac{2}{N\phi_0} = 2\beta_{\text{opt}}. \quad (11.72)$$

This condition is considerably more stringent than the condition for convergence of the average coefficient vector (11.39). This implies that we could get a situation where the average coefficient vector is converging but the norm of the error vector is diverging. Since this is unacceptable, (11.72) is the most stringent condition.

Since convergence is exponential, it is instructive to define a time constant  $\tau$  as the number of samples required for the MSE to decrease by a factor of  $e^{-1}$ ,  $\gamma^\tau = 1/e$ ; solving for  $\tau$  we get, in the range of small  $\beta$ ,

$$\tau \approx \frac{1}{2\beta\phi_0} \quad (11.73)$$

with a shortest time constant corresponding to  $\beta_{\text{opt}}$  of  $\tau \approx N/2$ . The important conclusion here is that the best rate of convergence is dependent on the number of filter coefficients — the more coefficients, the longer it takes for the coefficients to converge. This is not surprising in view of the inevitable interaction of the coefficients. The more coefficients there are, the more "noise" is introduced into the adaptation of each coefficient by the simultaneous adaptation of the other coefficients.

Aside from the rate of convergence, the other parameter of interest is the asymptotic error in the filter coefficients after convergence. The stationary point in (11.69) is

$$E[\|\mathbf{q}_k\|^2] \rightarrow \frac{N\beta}{2 - N\beta\phi_0} \xi_{\min} \quad \text{as } k \rightarrow \infty. \quad (11.74)$$

This is plotted in Figure 11-9, where we see that the asymptotic error increases as the step size increases until it blows up at twice the optimal step size. At the optimal step size (optimal in terms of rate of convergence of MSE, not the asymptotic MSE), the error is

$$E[\|\mathbf{q}_k\|^2] \rightarrow \frac{1}{\phi_0} \xi_{\min}. \quad (11.75)$$

In view of (11.66), the asymptotic MSE is

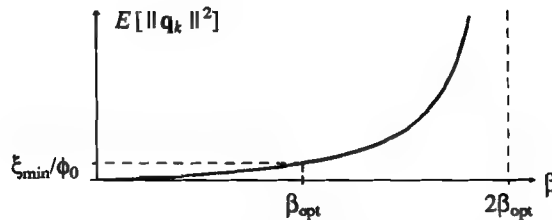


Figure 11-9. Asymptotic average Euclidean norm squared of the filter coefficients after convergence as a function of the step size.

$$E[|E_k|^2] \rightarrow \xi_{\min} + \xi_{\min} = 2\xi_{\min}. \quad (11.76)$$

Thus, for the fastest convergence, the total MSE is twice the minimum MSE for a fixed-coefficient filter, with half that MSE attributable to the asymptotic wandering of the filter coefficients about their optimal value.

There is an important tradeoff between speed of convergence and asymptotic MSE. If the goal is to minimize the asymptotic MSE, then choose as small a step size as possible. Generally what limits how small we can make the step size is the number of bits of precision in the arithmetic we use to implement the SG algorithm. In this case we can get the asymptotic MSE as close to the minimum MSE for a fixed-coefficient filter as we like at the expense of higher precision arithmetic. On the other hand, if our goal is to maximize the rate of convergence, then the step size should be chosen to be  $\beta_{\text{opt}}$ , with the penalty that the asymptotic MSE is twice as large as the minimum possible value.

A similar analysis to what we have done here can be applied to the general input spectrum case [8,9,10,11]. Surprisingly, the results are essentially the same as for the white input case we have considered here. Thus, the large effect of eigenvalue spread on the convergence of the average coefficient vector does not extend to the convergence of the MSE. The reason for this can be seen in the expression for excess MSE given in (11.42). In the direction of eigenvectors corresponding to small eigenvalues, the MSE does not change very rapidly, as manifested in the  $\lambda_i$  term. This implies that the coefficient vector tends to have larger excursions in this direction. Small eigenvalues thus cause problems in the convergence of the coefficient vector in the direction of the corresponding eigenvectors. However, these same excursions do not cause as large an impact on the resulting MSE precisely because of the small eigenvalue. Thus, if the goal is to accurately estimate the optimal coefficient vector, then small eigenvalues are a problem, but if the goal is to minimize the MSE of the adaptive filter they do not present nearly as great a problem. Even in the latter case, however, they can lead to numerical problems, as discussed in Section 11.4.

### 11.2.2. Common Modifications

Several modifications are commonly made to the SG algorithm as we have derived it.

#### Normalization of Step Size

The SG algorithm displays an undesirable dependence of speed of convergence on input signal power. This can be seen from (11.53); if the input signal is increased in size by a factor  $\gamma$ , then this is equivalent to increasing the step size  $\beta$  by the factor  $\gamma^2$  to  $\beta\gamma^2$ . Another way to see this is that the optimal step size of (11.71) requires that the step size  $\beta$  should be inversely proportional to the input signal power  $\phi_0$ . The speed of convergence and asymptotic MSE of the SG algorithm is strongly affected by the size of the input signal. A serious consequence is that if the input signal grows too large the adaptation algorithm becomes unstable.

The standard solution to this difficulty is to normalize the step size of the algorithm. The size of the updates can be kept approximately the same size on average if

the update is normalized by an estimate of the input signal power, which is equivalent to choosing a step size in (11.53) equal to

$$\beta_k = \frac{a}{\sigma_k^2 + b}, \quad (11.77)$$

where  $\beta_k$  is the step size at time  $k$ ,  $a$  and  $b$  are some appropriately chosen constants, and  $\sigma_k^2$  is an estimate of the input signal power at time  $k$ . The purpose of the  $b$  in the denominator is to prevent  $\beta_k$  from becoming too large (causing instability) when the input signal power becomes very small.

As an example of how the input signal power can be estimated, we can use an exponentially weighted time average of the input signal power,

$$\sigma_k^2 = (1 - \alpha) \sum_{j=0}^{\infty} \alpha^j |R_{k-j}|^2 \quad (11.78)$$

where  $\alpha$  is an appropriately chosen constant and the  $(1 - \alpha)$  factor normalizes the estimate to be an unbiased estimate of the input signal power. The reason for choosing this estimate is that it can be written recursively as

$$\sigma_k^2 = \alpha \sigma_{k-1}^2 + (1 - \alpha) |R_k|^2. \quad (11.79)$$

### Gear-Shift Algorithms

There is a tradeoff between asymptotic MSE and speed of convergence of the SG algorithm. Speed of convergence is important in two contexts. First, if we start the equalizer up on an unknown channel, then we would like to have rapid convergence of the equalizer. Second, if there is any variation of the channel, we would like the equalizer to track this variation. In many applications of adaptive equalization, variation of the channel is quite slow.

#### Example 11-5.

In voiceband data modems, there is no mechanism to cause any significant variation of the impulse response of the channel once a telephone connection is established. Thus, any changes are minor and occur over a long time period.  $\square$

On these channels, a very small step size would be desirable after convergence of the equalizer to insure a small asymptotic excess MSE, but a larger step size is needed during initial convergence. The solution is a *gear-shift* algorithm, in which the step size is initially larger, and shifted to a smaller value after a sufficient period of time for convergence to have occurred.

There are examples of channels in which the tracking capability of the equalizer is important, in which case a larger excess MSE must be accepted in order to gain this tracking capability.

#### Example 11-6.

In a microwave radio system, selective fading can vary fairly rapidly. Therefore, the time constant of the MSE adaptation should be small relative to the time of significant variation of the fading. In a mobile radio system, the variations are even faster, making speed of

adaptation the most important factor.  $\square$

### 11.3. ADAPTIVE DFE

Just as the coefficients of a transversal filter equalizer can be adapted, so too can the coefficients of a decision feedback equalizer. We will follow the model of the linear equalizer, and consider first the MSE solution for a finite precursor and postcursor equalizer filter in the DFE, followed by derivation of the stochastic gradient algorithm (we can dispense with the MSE gradient algorithm now that we know the principle of the stochastic gradient).

We will assume the form of the DFE shown in Figure 11-2 in which the precursor equalizer is anti-causal with  $N$  coefficients and the postcursor equalizer is causal with  $M$  coefficients. The slicer input is given by the relation

$$Q_k = \sum_{i=-(N-1)}^0 c_i R_{k-i} - \sum_{i=1}^M d_i \hat{A}_{k-i} \quad (11.80)$$

which is illustrated in Figure 11-10. The figure has been drawn to illustrate an interesting interpretation of the finite DFE as a two-sided transversal filter, with non-causal coefficients to cancel precursor intersymbol interference and causal coefficients to cancel postcursor intersymbol interference. This is similar to the LE, with the important difference that the input to the causal portion of the filter is the decisions rather than the output of the precursor equalizer filter. This difference will obviously change the desired tap coefficients as well as reduce the noise enhancement due to

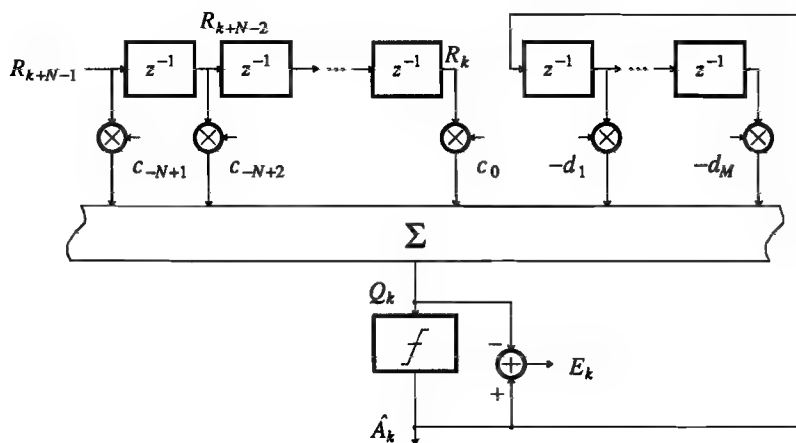


Figure 11-10. The DFE for finite precursor and postcursor equalizer transversal filters. In general all signals and coefficients are complex-valued.



equalization.

### Exercise 11-11.

The coefficients of *both* the causal and non-causal portions of the DFE equalizer will be different from the corresponding coefficients of the LE. Why?  $\square$

## 11.3.1. MSE Solution

It is instructive to find the optimal finite equalizers using the MSE criterion. We have to re-solve this problem, first considered in Chapter 10, since the filters now have constrained complexity. It is permissible to assume that there are no decision errors in calculating the output of the postcursor equalizer. The resulting filter, combining the channel model of Figure 11-2 and the equalizer of Figure 11-10, is shown in Figure 11-11a. After combining the filter blocks, we get the configuration of Figure 11-11b.

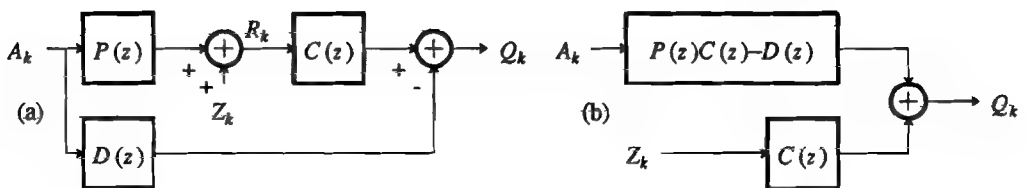
The objective is to minimize  $E[|E_k|^2]$  over the choice of the the postcursor equalizer filter coefficients and the precursor equalizer filter coefficients. The former are easier to find. Assuming the noise and data symbols to be uncorrelated,  $E[|E_k|^2]$  in Figure 11-11b is the sum of two terms, one for the noise and the other for the intersymbol interference, where the noise term is independent of  $D(z)$ . Hence, we can minimize just the intersymbol interference term over the postcursor equalizer coefficients. This term is of the form

$$\sum_m v_m A_{k-m} \quad (11.81)$$

where

$$v_k = \begin{cases} \sum_{i=-(N-1)}^0 c_i p_{k-i} - d_k, & 1 \leq k \leq M \\ \sum_{i=-(N-1)}^0 c_i p_{k-i}, & \text{otherwise} \end{cases} \quad (11.82)$$

It is evident that as long as the data symbols are uncorrelated with one another,  $E[|E_k|^2]$  will be minimized by choosing  $D(z)$  to eliminate the first  $M$  intersymbol interference samples; that is, force  $v_m = 0$ ,  $1 \leq m \leq M$ . Hence, the optimal postcursor



**Figure 11-11.** (a) Equivalent of channel, precursor equalizer, and postcursor equalizer assuming no decision errors. (b) Simplification after combining filters.

equalizer coefficients are

$$d_m = \sum_{i=-(N-1)}^0 c_i p_{m-i}, \quad 1 \leq m \leq M. \quad (11.83)$$

This says, not surprisingly, that there is no benefit to leaving any postcursor intersymbol interference after the postcursor equalization within the memory of the filter, since this cannot reduce the noise.

Having found the postcursor equalizer coefficients in terms of the precursor equalizer coefficients, we substitute this solution and then minimize over the precursor equalizer coefficients  $c_k$ .

#### Exercise 11-12.

Assume the data symbols are mutually uncorrelated with mean zero and variance  $\sigma_a^2$ . Define a vector of precursor equalizer coefficients

$$\mathbf{c}' = [c_{-(N-1)} \cdots c_0] \quad (11.84)$$

and show that the optimal  $\mathbf{c}$  satisfies (11.16) where the matrix  $\Phi$  has elements given by (11.11) except that the summation is missing the terms  $m = 1$  to  $m = M$  and the vector  $\alpha$  is given by

$$\alpha' = \sigma_a^2 [p_{-(N-1)} \cdots p_0]. \quad (11.85)$$

□

### 11.3.2. Stochastic Gradient Algorithm

The derivation of a stochastic gradient algorithm for the DFE is a simple extension of the LE case. First we define an augmented vector of  $N+M$  filter coefficients,

$$\mathbf{v}' = [c_{-(N-1)} \cdots c_0 \quad -d_1 \cdots -d_M] \quad (11.86)$$

and an augmented input signal vector

$$\mathbf{w}_k' = [R_{k+(N-1)} \cdots R_k \quad A_{k-1} \cdots A_{k-M}]. \quad (11.87)$$

Then the DFE slicer error can be expressed as

$$E_k = \hat{a}_k - \mathbf{v}_k' \mathbf{w}_k. \quad (11.88)$$

This is identical to the LE case with  $\mathbf{c}$  replaced by  $\mathbf{v}$  and  $\mathbf{p}$  replaced by  $\mathbf{w}$ , and hence we can immediately infer that the SG algorithm is, from (11.53),

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \beta E_k \mathbf{w}_k^*. \quad (11.89)$$

## 11.4. FRACTIONALLY SPACED EQUALIZER

Thus far in this chapter we have considered the adaptation of a linear equalizer with sample rate equal to the symbol rate. In practice it would be more common to adapt a *fractionally spaced equalizer (FSE)* discussed in Section 10.3 for several

reasons. First, incomplete knowledge of the channel makes it impossible to realize a matched filter directly. The FSE structure allows us, in effect, to adapt the matched filter as well as the equalizer. Second, the FSE is less influenced by sampling phase, as discussed in Section 10.3. Third, for implementation, the separate matched filter has the effect of doubling the loss of the channel, greatly increasing the gain necessary in the transversal filter equalizer and increasing its dynamic range requirements.

If we start with an FSE and use a MSE criterion for adaptation, within the constraints of the finite degrees of freedom the resulting filter will perform both the matched filtering and equalizer functions. The adaptation is also a straightforward extension of the earlier case. For example, if the FSE sampling rate is twice the symbol rate, we can think of the FSE as a transversal filter that operates at twice the symbol rate but simply fails to calculate every second output sample. The adaptation is based on the output samples that are calculated. However, the FSE does suffer from one subtle difficulty, which is a form of numerical ill-conditioning. We will address this issue in the following subsection.

### 11.4.1. Conditions for Unique MSE Solution

The existence of a unique solution to the MSE problem, as well as the arguments for the convergence of the gradient and SG algorithms, depended on the nonsingularity of the input autocorrelation matrix  $\Phi$ . The singular case corresponds to one or more zero eigenvalues. From the argument in Section 11.1, the case of concern is where the spectrum of the reference input vanishes at some frequency, since (11.47) would then predict that one (or more) eigenvalues would approach zero as  $N \rightarrow \infty$ . That the vanishing of the input spectrum would cause problems is not surprising, since the equalizer transfer function in the regions of zero spectrum does not affect the MSE, so the filter coefficients would obviously not be unique.

The FSE displays this ill-conditioning problem, because the bandwidth of the input data signal is deliberately made less than half the sampling rate. Although there will likely be noise components at all frequencies, they may be small and still not prevent some eigenvalues from being very small. Thus, we pay a price for the reduced sensitivity to sampling phase, and other benefits of the FSE, in adversely affecting the convergence properties of the equalizer. The small eigenvalues lead in particular to a problem with coefficient saturation, as will now be detailed.

### 11.4.2. Coefficient Drift

In any implementation, analog or digital, there will be a maximum value that a filter coefficient can assume. As one or more eigenvalues get very small, it becomes more likely that this maximum value will be inadequate, and the proper operation of the filter comes into question. This point is illustrated in Figure 11-12. The region of allowed filter coefficients for a two-coefficient filter is usually a square centered at the origin as constrained by implementation considerations. As an eigenvalue approaches zero, the sensitivity of the mean-square error in the direction of the corresponding eigenvector to the filter coefficients becomes very small. Even after convergence of the filter coefficients to the optimum, there will continue to be a fluctuation of the filter coefficients about that optimum, with the adaptation algorithm continually

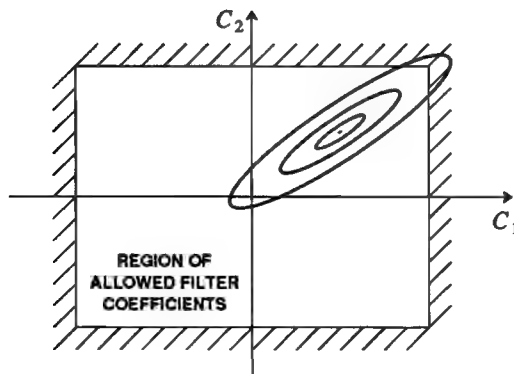
bringing the coefficients back toward the optimum. The fluctuation of the coefficients in the direction of least sensitivity (the direction of the eigenvector corresponding to the minimum eigenvalue) will tend to be larger. Since the coefficients can drift relatively freely in this direction, the phenomenon is called *coefficient drift*. As the eigenvalue gets smaller, the probability of coefficient drift taking the coefficients out of the allowed region gets large.

There are several possible solutions to this problem. If the coefficients drift to the edge of the allowed region, then further adaptation could cause an overflow. A simple solution is to saturate the coefficients when they reach the edge, in effect constraining the adaptation to the allowed region. A less attractive solution is to inject a small component of white noise at the input to the adaptation algorithm, thereby increasing the smallest eigenvalue (Problem 11-14). Obviously this degrades performance, since this noise component will also appear at the slicer input.

A third solution, which also degrades performance, is to introduce a *coefficient leakage* that tends to force the coefficients toward the origin [12]. This leakage can be obtained by changing the criterion that the adaptation algorithm is minimizing to

$$E[|E_k|^2] + \mu \|c\|^2, \quad (11.90)$$

where  $\mu$  is another small constant. Instead of simply minimizing the MSE, the criterion tries in addition to minimize the length of the coefficient vector. Like the added white noise, this criterion results in some compromise in the asymptotic mean-square error and coefficient vector, which are no longer optimal in the sense of (11.16) (Problem 11-15).



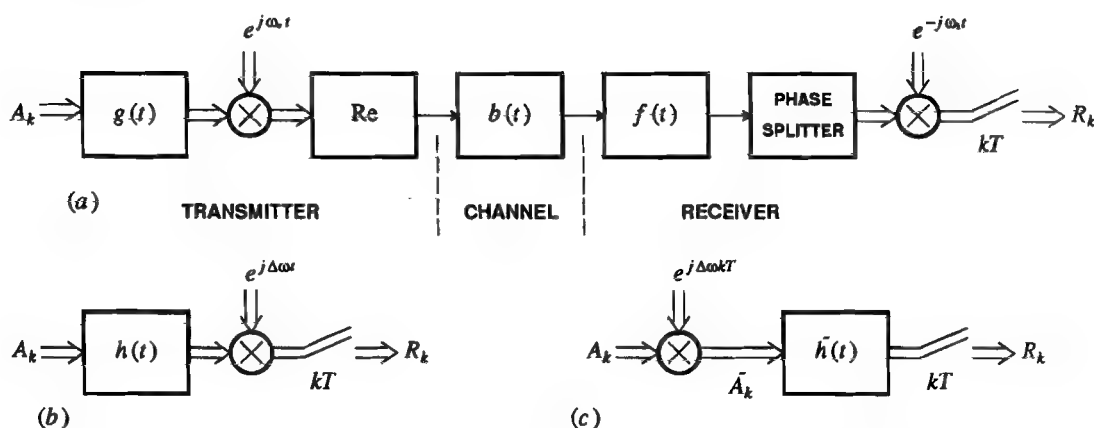
**Figure 11-12.** Contours of equal MSE for a large eigenvalue spread.

## 11.5. PASSBAND EQUALIZATION

The equalization techniques that we have discussed thus far are based on a baseband channel model, in which it has been assumed that demodulation has been performed prior to equalization. It is possible to perform equalization prior to demodulation; this is called *passband equalization*. This is an important extension of the results thus far. Passband equalization is much more common than baseband equalization, for reasons that will be elaborated in Chapter 16. Basically these reasons relate to the difficulty in placing the adaptive equalizer in the carrier recovery loop. Passband equalization mitigates these difficulties by allowing us to place the entire demodulation structure after the equalization. Passband equalization was proposed in a seminal paper by Falconer in 1976 [13]. Passband fractionally-spaced equalization was assumed in Figure 6-23.

The first step in deriving the passband equalizer structure is to derive a new channel model assuming that demodulation is not performed in the receiver front end. Such a channel model is shown in Figure 11-13a. We show the usual QAM transmitter, a channel impulse response  $b(t)$ , a receive filter  $f(t)$ , a phase splitter to generate the analytic signal, and a demodulator with frequency  $\omega_1$ . The baseband channel model that we have considered thus far corresponds to  $\omega_1 = \omega_c$ . The case where there is no demodulation corresponds to  $\omega_1 = 0$ . Other values of  $\omega_1$  are possible; an important example would be where  $\omega_1$  was chosen to *nominally* equal  $\omega_c$ , but with a small and unknown frequency offset due to the fact that  $\omega_c$  and  $\omega_1$  are generated by independent oscillators.

If we define



**Figure 11-13.** A passband channel model. (a) The transmitter, channel, and receiver assuming that demodulation uses frequency  $\omega_1$  rather than  $\omega_c$ . (b) Equivalent channel model consisting of baseband filter and modulator. (c) Equivalent channel model consisting of modulator and passband filter.

$$\Delta\omega = \omega_c - \omega_1 \quad (11.91)$$

then the channel model of Figure 11-13b can be derived.

**Exercise 11-13.**

Show that the model of Figure 11-13b follows from Figure 11-13a, where

$$h(t) = g(t) * ((b(t) * f(t))e^{-j\omega_c t}). \quad (11.92)$$

□

By a simple manipulation, the model of Figure 11-13b becomes

$$\sum_k A_k h(t - kT) e^{j\Delta\omega t} = \sum_k \tilde{A}_k \tilde{h}(t - kT) \quad (11.93)$$

where

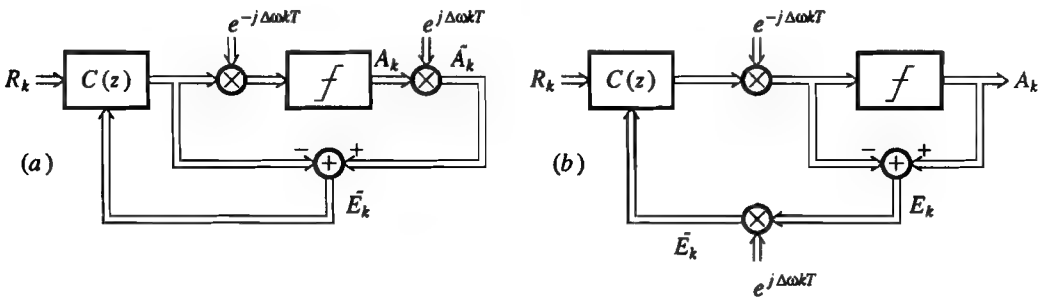
$$\tilde{A}_k = A_k e^{j\Delta\omega kT} \quad (11.94)$$

is called the *rotated data symbol* and

$$\tilde{h}(t) = h(t) e^{j\Delta\omega t} \quad (11.95)$$

is called the *passband channel response*. This relation corresponds to the model shown in Figure 11-13c. Since  $h(t)$  is a baseband filter, centered at d.c.,  $\tilde{h}(t)$  is a passband filter, centered at frequency  $\Delta\omega$ . One way to view this model is as a modulation operation followed by passband filter, as opposed to a baseband filter followed by modulation in Figure 11-13b.

For purposes of equalization, we can think of the channel as consisting of a passband filter  $\tilde{h}(t)$  driven by the rotated data symbols  $\tilde{A}_k$ . This logic leads to the passband equalizer structure shown in Figure 11-14a. The sampled channel output is fed through a passband equalizer  $C(z)$ , the purpose of which is to invert the response  $\tilde{h}(t)$  to yield a good estimate of the rotated symbols  $\tilde{A}_k$ . If we wanted to replicate the baseband equalizer structure, we would build a slicer appropriate for the rotated symbols. The simplest way to do this is to take a slicer appropriate for the non-rotated symbols, and precede that slicer by the reverse rotation and follow it by the rotation,



**Figure 11-14.** Passband equalizer structure. a. Direct generation of rotated error signal. b. Rotation of slicer error signal.

as shown in the figure. Since the purpose of the passband equalizer is to generate a good estimate of the rotated symbols, we must use an error signal for adaptation which is the difference between the equalizer output and the rotated data symbol, as shown. We call this error signal  $\tilde{E}_k$ , since it is a rotated version of the error  $E_k$  between input and output of the non-rotated slicer,

$$\tilde{E}_k = e^{j\Delta\omega kT} E_k. \quad (11.96)$$

This is made clearer by the equivalent equalizer structure shown in Figure 11-14b. We can think of this structure as realizing a slicer and error generator appropriate for the non-rotated data symbols, whereas the equalizer works in a rotated data symbol world. The rotators simply convert between the two worlds.

The convergence of the adaptive passband equalizer follows directly from the baseband case, since the two are equivalent except for two facts:

- The passband equalizer is driven by rotated data symbols rather than the non-rotated symbols.
- The passband equalizer is inverting the passband channel response  $\tilde{h}(t)$  rather than the baseband response  $h(t)$ .

The relationship between the statistics of the non-rotated and rotated data symbols is easily developed.

#### Exercise 11-14.

Show that the power spectrum of the rotated data symbols is given by

$$S_{\tilde{A}}(e^{j\omega T}) = S_A(e^{j(\omega-\Delta\omega)T}). \quad (11.97)$$

Hence, if the non-rotated symbols are white (uncorrelated), then so are the rotated symbols.  $\square$

The convergence properties of the passband equalizer are therefore the same as those of the baseband equalizer when the transmitted data symbols are uncorrelated.

The passband equalizer can be used in several ways:

- We can phase-lock the demodulation carrier to the incoming carrier, making  $\omega_1 = \omega_c$  ( $\Delta\omega = 0$ ). This raises problems to be discussed in Chapter 16, because it puts the equalizer delay into the carrier recovery loop, slowing the tracking capability of that loop.
- We can choose  $\omega_1 = 0$  ( $\Delta\omega = \omega_c$ ). This is the passband equalizer case.
- We can choose  $\omega_1$  to nominally equal  $\omega_c$ , but without phase locking with a carrier recovery loop. For this case  $\Delta\omega$  will be small but unknown. We can think of this case as a baseband equalizer with a small frequency offset and phase compensation at the output. In this case  $\tilde{h}(t)$  is not a passband function, although it has been shifted in frequency by a small amount.

In either of the second two cases, a carrier recovery loop driving the rotators at the equalizer output is required, as discussed in Chapter 16. Case 3 could therefore be described as a baseband equalizer with a phase-tracking carrier loop. The relative merits of the second two realizations is considered in Problem 11-17.

## 11.6. FURTHER READING

The tutorial article by Qureshi on adaptive equalization is highly recommended reading [10,11], as is the recent book by Proakis [14]. The early treatise by Lucky, Salz, and Weldon is somewhat dated but still recommended reading [3]. There are several books on the general topic of adaptive filtering [9,15,16].

We have not discussed the adaptation of a ML sequence detector (Viterbi algorithm). This is not straightforward; the issues are addressed at some length in [11].

Several methods to speed the convergence of the adaptive equalizers have been omitted. These include an alternative structure called the *lattice filter* [9,17,18], and a class of adaptation algorithms called the *least-squares (LS)* algorithms [9]. There are many versions of the LS algorithms, including those based on both transversal filter [19] and lattice filter realizations [20].

### APPENDIX 11-A SG ALGORITHM ERROR VECTOR NORM

In this appendix we approximate the expected value of the norm of the error vector  $\mathbf{q}_k$  for an input random process consisting of zero-mean independent samples.

An update for  $\mathbf{q}_k$  is given in (11.60). By direct calculation,

$$\begin{aligned} \|\mathbf{q}_{k+1}\|^2 &= \mathbf{q}_{k+1}^* \mathbf{q}_{k+1} \\ &= \mathbf{q}_k^* \Gamma_k^* \Gamma_k \mathbf{q}_k + 2\beta \operatorname{Re}\{D_k^* \mathbf{r}_k^* \Gamma_k \mathbf{q}_k\} + \beta^2 |D_k|^2 \|\mathbf{r}_k\|^2. \end{aligned} \quad (11.98)$$

Since by assumption  $D_k$  is the error of an optimal fixed-coefficient equalizer, by the orthogonality principle it is uncorrelated with the vector of input samples  $\mathbf{r}_k$ . If we also assume it is independent,

$$E[D_k^* \mathbf{r}_k^* \Gamma_k \mathbf{q}_k] = E[D_k^*] E[\mathbf{r}_k^* \Gamma_k \mathbf{q}_k] \quad (11.99)$$

and noting that  $E[D_k] = 0$ , fortuitously the expectation of the middle term is zero.

Assuming that the error vector  $\mathbf{q}_k$  is changing very slowly, we can assume it is a constant with respect to the expectation over the input random vector  $\mathbf{r}_k$ , and thus the mean value of the error vector norm versus time is

$$\|\mathbf{q}_{k+1}\|^2 = \mathbf{q}_k^* E[\Gamma_k^* \Gamma_k] \mathbf{q}_k + \beta^2 E[|D_k|^2] E[\|\mathbf{r}_k\|^2]. \quad (11.100)$$

In this expectation, we have

$$E[|D_k|^2] = \xi_{\min}, \quad E[\|\mathbf{r}_k\|^2] = N\phi_0. \quad (11.101)$$

To evaluate the expectation of the first term, we write it out explicitly,



$$\begin{aligned}
 E[\Gamma_k^{*'} \Gamma_k] &= \mathbf{I} - 2\beta\Phi + \beta^2 E[(\mathbf{r}_k' \mathbf{r}_k^*)(\mathbf{r}_k^* \mathbf{r}_k')] \\
 &= \mathbf{I} - 2\beta\Phi + \beta^2 E[\|\mathbf{r}_k\|^2 (\mathbf{r}_k^* \mathbf{r}_k')]
 \end{aligned}
 \quad (11.102)$$

where  $\Phi = \phi_0 \mathbf{I}$  by assumption. The last term is difficult since it involves fourth-order statistics, which we will have to approximate (unless the  $R_k$  are Gaussian, in which case exact evaluation is possible). In terms of the original input process, the  $m, n$  element of this matrix (indexed from the center) is

$$E\left[\sum_{j=-L}^L |R_{k+j}|^2 R_{k+m}^* R_{k+n}\right]. \quad (11.103)$$

Because of the assumed independence and zero mean of the  $R_k$ , this expectation is zero for  $m \neq n$ ; therefore, the matrix is diagonal. When  $m = n$ , it reduces to

$$\begin{aligned}
 E\left[\sum_{j=-L}^L |R_{k+j}|^2 |R_{k+m}|^2\right] &= E[|R_{k+m}|^4] + \sum_{\substack{j=1 \\ j \neq m}} E[|R_{k+j}|^2] E[|R_{k+m}|^2] \\
 &= \eta_a + (N-1)\phi_0^2
 \end{aligned}
 \quad (11.104)$$

where

$$\eta_a = E[|R_{k+m}|^4]. \quad (11.105)$$

We will approximate  $\eta_a$  as the square of the second moment,

$$\eta_a \approx \phi_0^2. \quad (11.106)$$

The second term is precisely  $(N-1)\phi_0^2$ ; hence, the entire sum is approximately  $N\phi_0^2$ . We can get some idea of the accuracy of this approximation from considering the Gaussian example.

#### Exercise 11-15.

Assume that  $R_k$  is a complex-valued Gaussian random variable with independent identically-distributed real and imaginary parts. Show that

$$\eta_a = 2\phi_0^2. \quad (11.107)$$

Hence the approximation above has the correct dependence on  $\phi_0$  but is off by a factor of two.  $\square$

Substituting the approximation of (11.106) into (11.104), we get

$$E[\Gamma_k^{*'} \Gamma_k] = (1 - 2\beta\phi_0 + \beta^2 N \phi_0^2) \mathbf{I} \quad (11.108)$$

and the result of (11.69) is established. The error in this approximation will generally be small when  $N$  is reasonably large. The approximation is necessary since the statistics are governed by the very complicated intersymbol interference, and the fourth moment is therefore very difficult to evaluate explicitly.

## PROBLEMS

- 11-1. For a linear predictor of input process  $R_k$ , define a vector of prediction coefficients

$$\mathbf{f}' = [f_1 \cdots f_N] \quad (11.109)$$

and a vector of past data samples,

$$\mathbf{r}_k' = [R_{k-1} \cdots R_{k-N}]. \quad (11.110)$$

Then the prediction error of an  $N$ -th order predictor is

$$E_k = R_k - \mathbf{f}' \mathbf{r}_k. \quad (11.111)$$

Find the optimal set of coefficients and the resultant minimum MSE.

- 11-2. Rederive (11.16) directly from orthogonality principle (11.26).  
 11-3. Derive the orthogonality principle for a linear predictor.  
 11-4. This problem will attempt to make plausible the relationship between the eigenvalues of an autocorrelation matrix and the power spectrum displayed in (11.47). Let  $\Phi$  be a  $(2L+1) \times (2L+1)$  autocorrelation matrix, and let the components of an eigenvector of this matrix be

$$\mathbf{v}' = [v_{-L}, v_{-L+1}, \dots, v_L]. \quad (11.112)$$

- (a) Show that the eigenvector and associated eigenvalue  $\lambda$  satisfies the relationship

$$\sum_{i=-L}^L \Phi_{j-i} v_i = \lambda v_j, \quad -L \leq j \leq L. \quad (11.113)$$

- (b) Let  $L \rightarrow \infty$  and take the Fourier Transform to show that

$$S(e^{j\omega T}) V(e^{j\omega T}) = \lambda V(e^{j\omega T}) \quad (11.114)$$

where  $S(e^{j\omega T})$  is the power spectrum defined by (11.46) and

$$V(e^{j\omega T}) = \sum_{i=-\infty}^{\infty} v_i e^{-j\omega T}. \quad (11.115)$$

- (c) Where  $S(e^{j\omega T})$  is a single valued function (that is, it doesn't assume the same value at two different frequencies), argue that the infinite eigenvectors have components that are samples of a complex exponential ( $e^{j\omega T}$ ), with corresponding eigenvalues equal to the power spectrum at the same frequency.  
 (d) Use these results to argue the validity of (11.47).  
 11-5. Consider an input wide-sense stationary random process with autocorrelation function  $\phi_k = \alpha^{|k|}$ .  
 (a) Find the power spectrum of this random process.  
 (b) Find the asymptotic minimum and maximum eigenvalues of the autocorrelation matrix.  
 (c) Find, as a function of  $\alpha$ , the eigenvalues and eigenvectors of the  $2 \times 2$  autocorrelation matrix.  
 (d) Find the eigenvalue spread of the autocorrelation matrix as predicted by approximate relation (11.47) and compare to the results of c.  
 (e) Find, as a function of  $\alpha$  and as  $N \rightarrow \infty$ , the step size  $\beta$  and resulting dominant mode of convergence of the MSEG algorithm. Interpret this result intuitively.  
 11-6. Show that for the MSEG algorithm, the error vector is given by

$$\mathbf{c}_j - \mathbf{c}_{\text{opt}} = \sum_{i=1}^n (1 - \beta \lambda_i)^j (\mathbf{v}_i^* (\mathbf{c}_0 - \mathbf{c}_{\text{opt}})) \mathbf{v}_i, \quad (11.116)$$

and interpret this equation.

- 11-7. Using the results of Problem 11-6 and (11.38) show that the excess MSE is given, for the MSEG algorithm, by the relation

$$E[E_k^2] - E[E_k^2]_{\min} = \sum_{i=1}^n \lambda_i (1 - \beta \lambda_i)^{2j} (\mathbf{v}_i^* (\mathbf{c}_0 - \mathbf{c}_{\text{opt}}))^2, \quad (11.117)$$

and interpret this equation.

- 11-8. Consider the dominant mode of the MSEG algorithm.
- The excess MSE as a function of time expressed in decibels is approximately given by  $\gamma_1 - \gamma_2 j$  and find the constants  $\gamma_1$  and  $\gamma_2$ .
  - Evaluate these constants for the particular case where  $\beta$  is very small, and discuss the tradeoff between speed of convergence and step size for this case. Thus, the MSE expressed in decibels decreases linearly with time.
- 11-9. For an input process  $Y_k$  with mean value  $\mu \neq 0$ , show that the minimum MSE first order predictor is

$$\hat{Y}_k = \rho Y_{k-1} + (1-\rho)\mu \quad (11.118)$$

where  $\rho$  is a normalized covariance defined as

$$\rho = \frac{\phi_1 - \mu^2}{\phi_0 - \mu^2}. \quad (11.119)$$

- 11-10. A real-valued input WSS process has power spectral density

$$\Phi(z) = \frac{A}{(1 - \alpha z)(1 - \alpha z^{-1})}, \quad 0 < \alpha < 1 \quad (11.120)$$

and the region of convergence includes the unit circle.

- Find the autocorrelation function  $\phi_k$ .
  - Find the predictor coefficients for a minimum MSE  $N$ -th order predictor.
- 11-11. In the MSEG algorithm, in place of a fixed step size  $\beta$ , use a variable step size  $\beta_j$  in the determination of  $\mathbf{c}_j$ .
- Show that the error vector at iteration  $j$  is given by

$$\mathbf{q}_j = \sum_{l=1}^n \prod_{i=1}^j (1 - \beta_i \lambda_i) (\mathbf{v}_i^* \mathbf{q}_0) \mathbf{v}_i. \quad (11.121)$$

- Show that you can force the error vector to zero in precisely  $N$  iterations by proper choice of the step sizes assuming you know the eigenvalues of the matrix (but that is all you need to know).
- 11-12. For the signal power estimation algorithm of (11.78), assume that  $R_k$  is a real-valued zero-mean white Gaussian process.
- Show that  $\sigma_k^2$  is an unbiased estimator of the signal power.
  - Find the variance of this estimate. Interpret how this variance depends on step size  $\alpha$ .
- 11-13. Assume an FSE with sampling rate equal to twice the symbol rate.
- Write the equations describing the input-output relationship of the FSE.
  - Find the SG algorithm that adapts this equalizer structure.

- 11-14.** Assume that a white noise component with variance  $\sigma^2$  is added to the input of an adaptive filter. Quantify the effect of this noise on the coefficient drift as follows:
- What is the new set of eigenvalues?
  - What is the new eigenvalue spread? What is the effect of  $\sigma$ ?
  - For the MSE solution, what is the effect of  $\sigma$  on the MSE?
- 11-15.** Suppose the MSE criterion is modified to minimize (11.90). Find the coefficient vector  $\mathbf{c}_\mu$  which minimizes this quantity, and show that the error between this solution and the coefficient vector  $\mathbf{c}_{\text{opt}}$  of (11.16) is given by

$$\mathbf{c}_\mu - \mathbf{c}_{\text{opt}} = -\mu \sum_{i=1}^n \frac{\mathbf{v}_i^* \alpha}{\lambda_i (\lambda_i + \mu)} \mathbf{v}_i, \quad (11.122)$$

and that the resulting excess MSE of (11.8) is given by

$$E[E_k^2] - E[E_k^2]_{\min} = \mu^2 \sum_{i=1}^n \frac{|\mathbf{v}_i^* \alpha|^2}{\lambda_i (\lambda_i + \mu)^2}. \quad (11.123)$$

How does this MSE increase as we vary  $\mu$ ?

- 11-16.** Continuing Problem 11-15:

- Find the MSE gradient algorithm which iteratively minimizes this error.
- Find the criterion on the step size of this algorithm which guarantees stability.
- Find the step size which maximizes the rate of convergence.
- Investigate how the maximum rate of convergence can be altered by the choice of  $\mu$ , particularly where the eigenvalue spread is large. Discuss the tradeoff between "excess MSE" and rate of convergence.
- How do these results apply to the stochastic gradient algorithm of (11.53)?

- 11-17.** Consider a voiceband data modem with the following characteristics: carrier frequency 1800 Hz, symbol rate 2400 Hz, excess bandwidth 10%. Assume that all sampling rates used in the receiver are an integer multiple of the symbol rate and are chosen to be as small as possible. Draw a block diagram of a receiver using a fractionally-spaced linear equalizer, labeling the sampling rates at each point.

- Assume a baseband equalizer with phase-locked demodulation at the receiver front end.
- Assume a passband equalizer.
- Assume a baseband equalizer with a demodulator at the front end, but not phase-locked to the receive carrier.
- Which of these realizations appears to be more attractive? Why?

- 11-18.** For the same conditions as in Problem 11-17b, draw a block diagram of a DFE with a passband precursor equalizer labeling the sampling rates at each point.

## REFERENCES

1. R. W. Lucky, "Automatic Equalization for Digital Communications," *BSTJ* **44** pp. 547-588 (Apr. 1965).
2. R. W. Lucky and H. R. Rudin, "An Automatic Equalizer for General-Purpose Communication Channels," *Bell Sys. Tech. J.* **46** p. 2179 (Nov. 1967).
3. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., *Principles of Data Communication*, McGraw-Hill Book Co., New York (1968).
4. B. Widrow and M. E. Hoff, Jr., "Adaptive Switching Circuits," *IRE WESCON Conf. Rec.*, pp. 96-104 (1960).
5. R. M. Gray, "On the Asymptotic Eigenvalue Distribution of Toeplitz Matrices," *IEEE Trans. on Information Theory* **IT-18** pp. 725-730 (Nov. 1972).
6. U. Grenander and G. Szego, *Toeplitz Forms and Their Applications*, University of California Press (1958).
7. R. Bellman, *Introduction to Matrix Analysis*, McGraw-Hill, New York (1960).
8. G. Ungerboeck, "Theory on the Speed of Convergence in Adaptive Equalizers for Digital Communication," *IBM J. Res. and Develop.*, pp. 546-555 (Nov. 1972).
9. M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*, Kluwer Academic Publishers, Boston (1984).
10. S. U. H. Qureshi and G. D. Forney, Jr., "Performance Properties of a T/2 Equalizer," *NTC '77 Proceedings*, O.
11. S.U.H.Qureshi, "Adaptive Equalization," pp. 640 in *Advanced Digital Communications Systems and Signal Processing Techniques*, ed. K. Feher, Prentice-Hall, Englewood Cliffs, N.J. (1987).
12. R. D. Gitlin, H. C. Meadors, Jr., and S. B. Weinstein, "The Tap-Leakage Algorithm: An Algorithm for the Stable Operation of a Digital ly Implemented, Fractionally Spaced Adaptive Equalizer," *BSTJ* **61**(8)(Oct. 1982).
13. D. D. Falconer, "Jointly Adaptive Equalization and Carrier Recovery in Two-Dimensional Digital Communication Systems," *BSTJ* **55**(3)(March 1976).
14. J. G. Proakis, *Digital Communications, Second Edition*, McGraw-Hill Book Co., New York (1989).
15. C. F. N. Cowan and P. M. Grant, *Adaptive Filters*, Prentice-Hall, Englewood Cliffs, New Jersey (1985).
16. B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, N.J. (1985).
17. J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction," *IEEE Trans. on ASSP* **ASSP-25** pp. 423-428 (Oct. 1977).
18. E. H. Satorius and S. T. Alexander, "Channel Equalization Using Adaptive Lattice Algorithms," *IEEE Trans. on Communications* **COM-27** p. 899 (Jun. 1979).
19. D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization," *IEEE Trans. on Communications* **COM-26** pp. 1439-1446 (Oct. 1978).
20. B. Friedlander, "Lattice Filters for Adaptive Processing," *Proceedings of the IEEE* **70**(8)O.

# 12

---

## SPECTRUM CONTROL

---

*Coding*, which refers to the translation between the user-provided information bits (source bits) and the transmitted data symbols (coded symbols), is discussed in this and the following two chapters. This chapter discusses the use of coding to control the statistics of the data symbols, thereby introducing a measure of control over the spectrum of the transmitted signal. For example, undesired correlations among information bits, can be removed by *scrambling* (Section 12.5), which is a reversible transformation of the bits in a way that affects the statistics. Alternatively, the spectrum can be controlled by introducing a controlled correlation among data symbols in the form of *redundancy* (the remaining sections). In Chapters 13 and 14 we will see applications of redundancy to the correction and prevention of channel errors.

One way to control the spectrum is through the design of a *line code* (Sections 12.2 and 12.3). One major motivation in baseband systems is the problem of *baseline wander* introduced by the a.c. coupling inherent in transformers and broadband amplifiers. This phenomenon is described in Section 12.1. In Section 12.2 a number of different types of line codes for baseband systems are described, most of them oriented toward control of baseline wander. Then in Section 12.3, we describe a different class of techniques based on introducing spectral nulls at arbitrary frequencies using filtering. The resulting ISI is mitigated using transmitter precoding (Chapter 10). An important special case is called *partial response*. A related class of techniques that achieve high bandwidth efficiency and immunity to nonlinearities in bandpass systems, called *continuous-phase modulation*, is described in Section 12.4.

Often we want to ensure that the transmitted signal has sufficient *randomness* or activity so that timing recovery and other functions can be reliably performed. For example, because people type relatively slowly, a computer terminal transmits null characters most of the time. A long sequence of null characters results in a highly correlated line signal. Such signals can foil timing recovery (Chapter 17), adaptive equalization (Chapter 11) and echo cancellation (Chapter 19), all of which assume that the transmitted symbols are uncorrelated. *Scrambling*, described in Section 12.5, is intended to *remove* strong correlations among the information bits so as to make them appear more random.

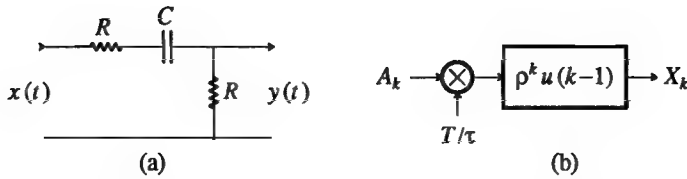
## 12.1. GOALS OF LINE CODES

In Chapter 6 we saw examples of different signaling schemes, such as binary antipodal and orthogonal signaling. These are simple examples of line codes. Line codes can be used to accomplish desirable goals such as to place spectral nulls at particular frequencies. A common goal is to introduce a null in the spectrum at d.c., thereby enabling transmission of a baseband PAM waveform over a channel that cannot accommodate a d.c. component in the data signal. An alternative for such a channel would be to use passband PAM, but in many applications baseband PAM in conjunction with line coding is a more cost-effective alternative.

Line coding is not as big an issue in passband systems as baseband, for several reasons. Baseband systems, particularly those operating over cable, typically have a large variation in attenuation over the Nyquist bandwidth and also a large variation in crosstalk coupling loss (Section 5.2). Hence, there is much that can be done to improve the performance of these systems by control of the transmitted power spectrum. Passband systems, in contrast, usually have a relatively constant attenuation vs. frequency (since the bandwidth is narrow relative to the center frequency), and crosstalk coupling may not be an issue or is relatively frequency-independent. In this chapter we will therefore limit our discussion of line codes to baseband PAM systems.

A common consideration in the line code is the tradeoff between symbol rate and the number of transmitted levels. As discussed in Chapter 6, the symbol rate relates directly to the required channel bandwidth, while the number of levels relates directly to the noise immunity. The line code also affects the transmitted power spectrum and hence the crosstalk into foreign systems (as in wire-pair or radio transmission) and radio-frequency interference (RFI). The line code affects many aspects of the implementation, such as the complexity of the equalization, detection, echo cancellation, and timing recovery circuitry.

A side benefit of some line codes is the ability to perform *in-service monitoring* of the data signal to be sure that the system is performing well even while transmitting the information bits. The coder adds redundancy to the transmitted data symbols, and the receiver checks to see that the code constraints are preserved after detection. Line coding can also be used to make the digital transmission signal more immune to nonlinearities, such as those on satellite and radio channels (Section 12.4).



**Figure 12-1.** a. An a.c. coupled circuit. b. A discrete-time system characterizing the postcursor ISI introduced by baseline wander in this circuit.

Important properties of line codes include their redundancy, their running digital sum, and their power spectrum.

### 12.1.1. Redundancy

If the number of distinct transmitted symbols is  $L$ , and the symbol rate is  $f_b$  symbols per second, then the information-carrying capacity is

$$R = f_b \log_2 L \quad \text{bits/sec.} \quad (12.1)$$

Define  $B$  as the information bit rate provided to the user. When  $B = R$ , there is no redundancy in the code, and our only degree of freedom in the design of the code is the choice of the deterministic translation between information bits and transmitted data symbols, described in detail in Chapter 6.

#### Example 12-1.

If  $L = 4$ , without redundancy we can assign two information bits to each data symbol. There are  $4! = 24$  possible ways in which we can assign these two bits to the four distinct data symbols.  $\square$

For many practical line codes,  $B < R$ . The difference between  $B$  and  $R$  represents a *redundancy* that can be put to good use. If the information bits are statistically independent, and  $B = R$ , then the transmitted data symbols must also be independent. But by allowing redundancy, we can make the transmitted data symbols statistically dependent, regardless of the statistics of the information, and hence exercise some control over the power spectrum of the transmitted signal.

### 12.1.2. Running Digital Sum

Many baseband systems use transformer coupling or a.c.-coupled electronics, which implies that the channel has infinite loss at d.c. In a sense the channel is actually passband, although it is special in that the highpass cutoff frequency is small relative to the symbol rate. Line coding techniques can deal with this situation, in place of more complicated carrier modulation techniques. Line coding also allows us to concentrate the signal power at frequencies near d.c., where the cable attenuation is often the lowest. Actually, as we will see, some line coding techniques are actually closely related to carrier modulation.

The effect of a.c. coupling on a channel is a form of intersymbol interference (ISI) called *baseline wander*. This effect, for the a.c. coupling circuit in Figure 12-1a



(or of a transformer, which is similar), on an input PAM signal  $\sum_k A_k \delta(t - kT)$  is analyzed in Problem 12-1. The conclusion is that the equivalent discrete-time channel is approximately characterized by the equivalent system generating the postcursor ISI in Figure 12-1b, where  $\tau = 2RC$  is the time constant and  $\rho = e^{-T/\tau}$ . This undesired *baseline wander* ISI, a consequence of the zero at d.c. in the channel response, is a major consideration in the choice of a line code.

There are two things needed to minimize the baseline wander problem. First, we must make the time constant  $\tau$  large (or a.c. coupling cutoff frequency small), since this will minimize the  $T/\tau$  term. When  $\tau$  is large,  $\rho \approx 1$ , and the ISI at sample  $k$  is approximately

$$X_k \approx \frac{T}{\tau} S_{k-1}, \quad S_k = \sum_{m=-\infty}^k A_m, \quad (12.2)$$

where  $S_k$  is called the *running digital sum (RDS)*. The RDS is an important property of the line code [1] because it predicts accurately the magnitude of the baseline wander ISI for a low cutoff frequency. Further, it is easily shown [2] that when the RDS is bounded, there is a spectral null at d.c.; conversely, for spectra generated by finite-state machines with a null at d.c., the RDS is bounded [3]. The second design consideration for the line code is therefore to ensure that the RDS  $S_k$  is small. For a line code not explicitly designed to minimize baseline wander, the RDS could grow to infinity. In this case the ISI in Figure 12-1b would not actually grow to infinity (Problem 12-2) since  $\rho < 1$ , but it could become quite large.

### 12.1.3. Transmitted Power Spectrum

The transmitted power spectrum is given by (3.82), which we repeat here for convenience. The PAM signal given by

$$X(t) = \sum_{k=-\infty}^{\infty} A_k g(t - kT) \quad (12.3)$$

is not wide-sense stationary, even if it is assumed that the transmitted data symbols  $A_k$  are wide-sense stationary. For some purposes, however, it is permissible to randomize the phase epoch in (12.3); this yields wide-sense stationarity, and thus the power spectrum is (Appendix 3-A)

$$S_X(j\omega) = \frac{1}{T} |G(j\omega)|^2 S_A(e^{j\omega T}). \quad (12.4)$$

The line code (along with the statistics of the source data) determines  $S_A(e^{j\omega T})$ , and the pulse shape chosen influences the power spectrum through the  $|G(j\omega)|^2$  term.

## 12.2. LINE CODE OPTIONS

Dozens of line codes have been seriously proposed. This section gives a representative set of these codes [4,5].

### 12.2.1. Linear Line Codes

*Linear line codes* are those in which the transmitted data symbols depend linearly on the information bits. In this subsection we will discuss three such line codes: the *binary antipodal* code, the *twinned binary* code, and *alternate mark inversion (AMI)*.

#### Binary Antipodal Codes

The simplest line codes transmit a pulse or its negative to send a "zero" or "one" bit, respectively (binary antipodal signaling). Several commonly used pulse shapes for binary antipodal signaling are shown in (12.4).

The only way to ensure no d.c. content for all possible transmitted data symbol sequences is to choose a pulse shape with no d.c. content (that is,  $G(0) = 0$ ). The only pulse shape in (12.4) with this property is the *biphase* or *Manchester* pulse [6]. For the other two pulse shapes, RZ and NRZ, we must somehow insure that the average rates of positive and negative pulses are equal (we will see ways of doing this shortly). The use of biphase is simpler, but in spite of the zero in the spectrum at d.c. for this pulse, there will still be baseline wander due to the low-frequency attenuation of the a.c. coupling. In other words, while the integral of the pulse is zero, the convolution of the pulse with a decaying exponential will have small but non-zero area. The magnitude of this source of intersymbol interference is easily predicted for the biphase (or any other) pulse shape.

#### Exercise 12-1.

Define  $\beta$  as the ratio of the cutoff frequency of the a.c. coupling to the symbol rate. Show that for the biphase code, the intersymbol interference from the last transmitted symbol has magnitude

$$(1 - e^{-\pi\beta})^2. \quad (12.5)$$

Thus, as the cutoff frequency decreases, this intersymbol interference goes to zero.  $\square$

#### Example 12-2.

If we require that the intersymbol interference from the last symbol be down by a factor of  $10^{-2}$ , then  $\beta = 0.033$ ; i.e., the cutoff frequency must be 3.3% of the symbol rate.  $\square$

The biphase line code and a similar code known as the *Walz pulse shape* (Problem 12-3) are often chosen for their "self-equalizing" properties, meaning that a single

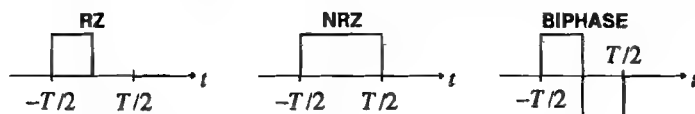


Figure 12-2. Return-to-zero (RZ), non-return-to-zero (NRZ), and biphase pulses.

compromise equalizer will suffice for a wide range of line lengths in wire-pair and coax systems. The intuitive reason for this is that since the response to a single positive-going pulse has a long tail, and  $g(t)$  consists of a positive-going pulse followed immediately by a negative-going pulse, the tail of the negative pulse will tend to cancel the tail from the positive pulse.

Choosing a pulse shape with zero integral is an effective and simple solution to the baseline wander problem. In addition, the fact that every symbol interval has a zero crossing in the center simplifies timing recovery (Chapter 17). However, we pay a high price for this zero crossing, since the high-frequency energy in the transmitted signal is larger. It is shown in Problem 12-4 that any pulse with no d.c. content that obeys the Nyquist criterion must have at least 100% excess bandwidth, or twice the minimum bandwidth of pulses that are allowed to have non-zero integral.

The biphasic pulse shape can be viewed in two alternative ways which will suggest other ways to eliminate the d.c. component of a data waveform.

- Start with a binary antipodal code with NRZ pulses, and multiply the resulting signal by a square wave with period equal to the symbol interval. We can think of this square wave as a carrier modulation, approximately centering the signal at the symbol rate. Since the bandwidth of the signal before modulation can easily be less than the symbol rate, this modulation avoids a d.c. content in the signal. Thus, biphasic can be viewed as a simple passband PAM scheme. This interpretation also explains why the resulting biphasic data signal bandwidth following equalization is roughly twice as great as for NRZ or RZ.
- Increase the symbol rate to twice the incoming bit rate, and use binary antipodal signaling with an NRZ pulse shape (at this higher symbol rate). Follow each information bit by another bit which is its complement; the result is a biphasic pulse shape. For example, information bits (...0011...) would become (...01011010...), at double the symbol rate. Thus, we can view biphasic as being equivalent to NRZ, with redundancy added by doubling the symbol rate.

The biphasic or similar pulse shape is a good choice where implementation simplicity is desirable and the distance between transmitter and receiver is modest, as in a local-area network (Chapter 1). But where we want to increase range by limiting the signal bandwidth, we will find better alternatives. These alternatives minimize baseline wander by more sophisticated means, and allow us to concentrate the signal power at lower frequencies where cable attenuation and crosstalk are lesser problems.

## Twinned Binary Code

We can improve on biphasic at the expense of additional implementation complexity by coding the transmitted data symbols. In other words, we can introduce a zero at d.c. in  $S_A(e^{j\omega T})$  rather than  $G(j\omega)$  by forcing the transmitted data symbols to be correlated. The only way that redundancy can be introduced without increasing the symbol rate is by increasing the number of levels. In a *pseudoternary line code*, we use a three-level data symbol to transmit one bit of information. The redundancy inherent in transmitting only one bit of information with three levels can be used to accomplish many goals, including the reduction of baseline wander. Whereas with the biphasic code we paid a price of greater signal bandwidth, with pseudoternary line

codes we suffer a reduction in noise immunity; i.e., for the same peak power level a smaller noise level will cause an error (Chapter 6).

An example of such a code is the *twinned binary code* invented by Meacham [7]. This code is not practical, but is easy to understand and can be made practical by a simple modification shown later. If the transmitted information bits are designated  $b_k$  and assume the values "0" and "1", then the transmitted symbols are

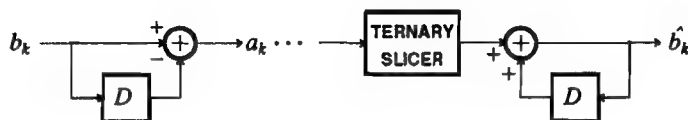
$$a_k = b_k - b_{k-1}. \quad (12.6)$$

It is easy to verify that  $a_k$  assumes the three values  $\{-1, 0, +1\}$ . For simplicity, throughout the remainder of this section we will denote the levels of a ternary code by "+", "0", and "-", with the obvious association to the actual pulse amplitudes. Because only one bit of information is conveyed per data symbol, the code is pseudoternary. In fact, we can make the following association: for a positive transition in the bit stream (from "0" to "1") transmit a "+", for a negative transition transmit a "-", and for no transition transmit a "0". From the power spectrum,

$$S_A(e^{j\omega T}) = S_B(e^{j\omega T}) |1 - e^{-j\omega T}|^2 = 4S_B(e^{j\omega T}) \sin^2\left(\frac{\omega T}{2}\right), \quad (12.7)$$

we see that a zero has been introduced into the spectrum at d.c. (and at all multiples of the symbol rate, since the spectrum is periodic in the symbol rate).

The twinned binary code is illustrated in Figure 12-3. We will denote the delay operator by  $D$  rather than  $z^{-1}$  in this and the following two chapters, as is conventional in the coding literature. In the receiver we have a ternary slicer; that is, a slicer appropriate for a ternary signal, with decision thresholds at  $\pm 1/2$ . The decoder, which follows the slicer, simply implements the transfer function  $(1-D)^{-1}$ , which is the reciprocal of the encoder transfer function  $(1-D)$ . The idea here is similar to the DFE in Chapter 10 — we introduce intersymbol interference in the coder, and eliminate it using past decisions in the decoder. There is a major difference however, in that the decoder follows the slicer (hard decoding) rather than surrounding the slicer as in the DFE, and thus we do not gain the noise immunity advantages of the DFE. This approach has the same problem as the DFE; namely, error propagation results if the encoder and decoder ever get into different states due to decision errors.



**Figure 12-3.** Coding and decoding for the twinned binary code.  $D$  is the delay operator, corresponding to  $z^{-1}$ .

**Example 12-3.**

Observe what happens if we have a long sequence of 1's followed by 0's on the input data stream. Then the input and output of the coder are shown below:

$$\begin{array}{r} 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\ 1 \ 0 \ 0 \ 0 \ 0 \ 0 -1 \ 0 \ 0 \ 0 \ 0 \end{array} \quad (12.8)$$

Notice that the only distinguishing characteristic between a sequence of 0's and 1's is the polarity of the coder output at the *beginning* of the sequence. In the receiver, a single decision error at that point can easily cause a sequence of 1's to be turned into a sequence of 0's, or *vice versa*.  $\square$

Actually the error propagation displayed in (12.8) is fundamental to the  $(1-D)$  response of the channel. Because this channel response passes no d.c., strings of inputs of the same polarity tend to be confused at the channel output. We saw this same phenomenon for the ML sequence detector on the same channel in Example 9-38. In that case, the sequence detector had a infinite number of minimum-distance error events corresponding to sequences of inputs of the same polarity. We will see a simple solution to this problem in the next subsection.

The redundancy inherent in transmitting one bit of information per ternary symbol can be used for in-service monitoring. This is a major advantage shared by all pseudoternary codes. All combinations of ternary digits are not possible; any impossible combination at the slicer output indicates that an error has been made. For example, two "+"s in a row, even with an arbitrary number of intervening 0's, is impossible because this would indicate two positive transitions in a row in the input bit stream  $b_k$ . In fact, we can state the redundancy constraint rather concisely. Any number of 0's in a row are allowed; every non-zero symbol must have the opposite polarity from the last non-zero symbol. This latter property ensures that there is no d.c. content in the transmitted signal. It also follows that the RDS is bounded by either  $0 \leq S_k \leq 1$  or  $-1 \leq S_k \leq 0$ , depending on the polarity of the first non-zero pulse transmitted. This implies that the *digital sum variation (DSV)*, the difference between the largest and smallest RDS, is unity for this line code. This is the smallest possible DSV for a pseudoternary code.

It is important to note the distinction between the biphase code and the twinned binary code. The former introduces a zero at d.c. by changing the transmitted pulse shape, and in the process, boosting the high frequencies in the signal. The transmitted signal still has two possible levels at any time. In the twinned binary code, we are able to introduce the zero without changing the transmitted pulse, but the price we pay is an increase in the number of levels to three, a need for a ternary or three-level slicer, and reduced noise immunity for the same peak transmitted power.

We can improve the spectral properties of the twinned binary code by *two-way interleaving*. To do this, replace (12.6) by

$$a_k = b_k - b_{k-2} \quad (12.9)$$

**Exercise 12-2.**

Show that the power spectrum of (12.9) has a null at both d.c. and at half the symbol rate,  $\omega = \pi/T$ .  $\square$

The advantage of having a null in the spectrum at half the symbol rate is that it can make practical a system with zero excess bandwidth without the requirement for "brickwall" lowpass filters. We will elaborate on this point in Section 12.3. But for our present purpose, it is valuable to understand another interpretation of (12.9), namely as two interleaved and independent twinned binary encoders. This interpretation follows from rewriting (12.9) as separate equations, one for even and one for odd-numbered data symbols,

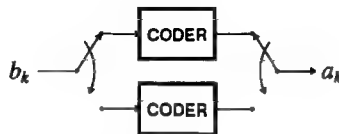
$$a_{2k} = b_{2k} - b_{2(k-1)}, \quad a_{2k+1} = b_{2k+1} - b_{2(k-1)+1}. \quad (12.10)$$

Observe that these two equations are independent, since one involves only even-numbered input bits and the other odd-numbered bits. This leads to the interpretation of Figure 12-4. The input bit stream is decimated to two half-rate streams, and each is applied to its own line coder. For our present example, each line coder happens to be a twinned binary coder.

The interleaved configuration of Figure 12-4 remarkably always leads to a spectral null at half the symbol rate if the constituent line coders have spectral nulls at d.c. This is because whenever the transmitted data symbols have a spectral null at d.c., they must also have a spectral null at the symbol rate (because of the periodicity of the sampled-data power spectrum). Since in Figure 12-4 each coder is operating at half the symbol rate, the spectrum of each must have a null at half the symbol rate. The superposition of their outputs preserves that property! This implies that we can get a half-symbol rate spectral null starting with any favorite line code simply by interleaving, although we will always pay the price of doubling of the RDS, with a resultant increase in baseline wander.

**Exercise 12-3.**

Argue that the RDS of the interleaved line coders of Figure 12-4 will be double the RDS of the constituent line code. Hence, for the twinned binary code, the RDS will fall in the range  $-2 \leq \text{RDS} \leq 2$ . What is the DSV?  $\square$



**Figure 12-4.** Two interleaved line coders.

## AMI Line Code

The error propagation problem of the twinned binary code can be overcome using *precoding*, similar to the transmitter precoding of Section 10.1; the result is known as *alternate mark inversion (AMI)* or a *bipolar* line code. This line code was invented by Barker and is commercially and historically important because it was used in the first commercial PCM system, the T1-Carrier system designed by Bell Laboratories in 1962 [8].

Error propagation occurs in the twinned binary code because the  $(1 - D)$  response causes an ambiguity for long strings of input 0's or 1's. In particular, these two cases are easily confused with one another. We can remove this ambiguity by a form of *differential encoding* of the bit stream prior to application to the line coder. A precoder and postcoder are shown in Figure 12-5, where the symbol " $\oplus$ " has the special meaning of *modulo-two summation*. For binary inputs, the modulo-two summation is the same as an *exclusive-or* circuit, with truth table given below:

Inputs		Output
0	0	0
0	1	1
1	0	1
1	1	0

The precoder function can be represented by the equation

$$c_k = b_k \oplus c_{k-1} \quad (12.11)$$

where  $c_k$  is also a bit (assuming values "0" and "1").

The precoder and postcoder recover the original bit, since from Figure 12-5 the output is

$$b_k \oplus c_{k-1} \oplus c_{k-1} = b_k. \quad (12.12)$$

This strange looking result follows from

$$c_{k-1} \oplus c_{k-1} = 0 \quad (12.13)$$

as the reader can readily verify.

The complete AMI coder is shown in Figure 12-6; it is simply a combination of the twinned binary coder and the differential precoder. There are two different kinds

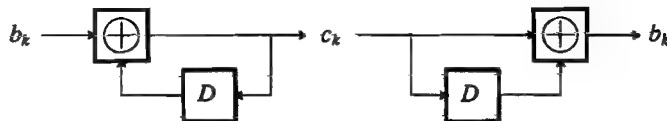


Figure 12-5. A differential precoder and postcoder for the AMI line code.

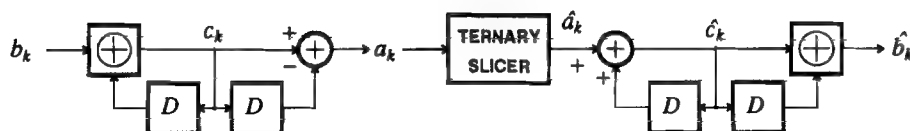


Figure 12-6. An AMI coder realized by precoder followed by  $(1 - D)$  filter.

of adders in this diagram, a normal adder and a modulo-two adder. For clarity, two distinct delays have been used where in fact one would suffice.

The precoder is called differential because it can be described as follows: a "zero" bit at the input is transmitted as no change in the precoder output bit, and a "one" bit is transmitted as a change in the output bit (either from one to zero or zero to one). This removes the ambiguity that caused the error propagation because sequences of 0's and 1's at the precoder output both correspond to 0's at the precoder input.

#### Example 12-4.

Repeating Example 12-3 with the precoder inserted, we get the following sequences at the input of the precoder, output of the precoder, and output of the  $(1 - D)$  filter:

$$\begin{array}{cccccccccc} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ + & - & + & - & + & - & 0 & 0 & 0 & 0 \end{array}$$

This assumes that the precoder output was initially zero. Strings of 1's are transformed into alternating plus-minus, which easily passes through the a.c. coupled channel.  $\square$

This precoding idea will be generalized in section 12.3, where we will see that AMI is a special case of *partial response coding*.

Another interpretation of the absence of error propagation is that the decoder function is *memoryless*, i.e. the current input bit can be determined from the ternary slicer output without regard to the past (equivalently, the decoder has no internal state). To verify the memoryless property, write the decoder mathematically as

$$\hat{c}_k = \hat{a}_k + \hat{c}_{k-1} \quad (12.14)$$

$$\hat{b}_k = \hat{c}_k \oplus \hat{c}_{k-1} = (\hat{a}_k + \hat{c}_{k-1}) \oplus \hat{c}_{k-1}. \quad (12.15)$$

It will turn out that in (12.15),  $\hat{b}_k$  is a function only of  $\hat{a}_k$  and not of  $\hat{c}_{k-1}$ . Consider for example the case  $\hat{a}_k = 1$ . Then for  $\hat{c}_{k-1} = 0$ ,

$$(1 + \hat{c}_{k-1}) \oplus \hat{c}_{k-1} = (1 + 0) \oplus 0 = 1 \oplus 0 = 1 \quad (12.16)$$

and similarly for  $\hat{c}_{k-1} = 1$ ,

$$(1 + \hat{c}_{k-1}) \oplus \hat{c}_{k-1} = (1 + 1) \oplus 1 = 2 \oplus 1 = 1. \quad (12.17)$$

Considering the other two cases, we can build the following truth table for the



decoder:

$a_k$	$\hat{b}_k$
+	1
0	0
-	1

To reiterate, the decoder output is a memoryless function of the slicer output, so the decoder has no internal state to get out of synchronization with the encoder, and there can be no error propagation. Based on this simplified descriptions of the encoder and decoder, we can state the AMI line code succinctly: encode a "zero" as a "0" transmitted symbol, and code a "one" alternately as "+" and "-". At the decoder, map a "0" received level into a "zero", and both "+" and "-" as "one".

**Exercise 12-4.**

Show that the following alternative description of the AMI coder is valid. The coder keeps track of the RDS  $s_k$ , which can only assume the values "0" and "+1", depending on whether the last non-zero pulse had negative or positive polarity respectively (this assumes that the first non-zero symbol was "+"). The coder then obeys the following truth table:

$b_k$	$s_{k-1}$	$a_k$	$s_k$
0		0	$s_{k-1}$
1	0	+	1
1	1	-	0

In this table the blank entry for  $s_{k-1}$  when  $b_k = 0$  means that we don't care what  $s_{k-1}$  is, and a "0" level is always transmitted.  $\square$

A different but equally useful generalization of AMI follows from another description. We can think of the AMI code as providing two mappings from  $b_k$  to  $a_k$  depending on the RDS at the last symbol as in the following table:

$s_{k-1} = 0$		$s_{k-1} = 1$	
$b_k$	$a_k$	$b_k$	$a_k$
0	0	0	0
1	+	1	-

The two mappings have the characteristic of being one-to-one (so that we can recover the data symbol at the receiver); one increases the RDS, while the other decreases it. We decide which mapping to use on the basis of the RDS at the last symbol, and in particular choose it to keep the RDS in the range [0,1]. A special class of line coders, called *sequence-state coders* generalize this idea (Section 12.2.2).

Calculating the power spectrum of the AMI code represents an interesting challenge, as we need to use the Markov chain results of Chapter 3. Interestingly, the AMI precoder is the same as the parity check circuit used as an example in Section 3.3 (see Figure 3-8). Let the input bit stream consist of independent bits, and let the probability of a "one" be  $p$ . We will find, not surprisingly, that  $p$  will have a large

influence on the power spectrum, which is determined for the precoder in Problem 3-29. The AMI coder output is the precoder output filtered by the linear time-invariant filter in Figure 12-6,

$$H(z) = 1 - z^{-1}. \quad (12.18)$$

The output power spectrum is therefore obtained by multiplying by  $H(z)H(z^{-1})$ ,

$$S_A(z) = \frac{p(1-p)(1-z^{-1})(1-z)}{(1-(1-2p)z^{-1})(1-(1-2p)z)}. \quad (12.19)$$

Evaluating on the unit circle we get the power spectrum

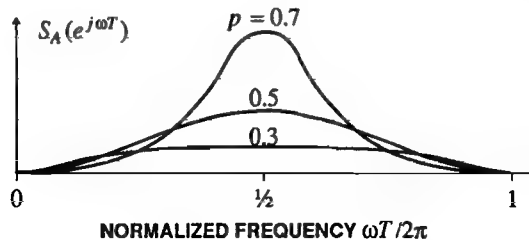
$$S_A(e^{j\omega T}) = 2p(1-p) \frac{1 - \cos \omega T}{1 + (1-2p)^2 - 2(1-2p) \cos \omega T}. \quad (12.20)$$

The power spectrum after pulse shaping can of course be found using (12.4). The power spectrum of (12.20) is plotted in Figure 12-7. Note the nulls in the spectrum at  $\omega = 0$ , as expected, and also at all multiples of the symbol rate  $\omega = 2\pi/T$ , due to the periodicity of the spectrum. Also note the influence of  $p$ , with large  $p$  resulting in a preponderance of power near half the symbol rate, due to the alternating "+" and "-" pulses. The number of transitions in the coded waveform is directly related to the density of "ones" in the information bit stream; for ease of timing recovery most systems that use AMI also place restrictions on the minimum density of "ones".

AMI shares with the twinned binary code one serious flaw: it is possible to have a long series of transmitted "0" levels, corresponding to a long series of "zero" inputs. This corresponds to a long period where the transmitted signal is zero, which can cause timing recovery circuits to lose synchronization (Chapter 17). This flaw was acceptable in the early days of PCM because the bit stream was always an encoded version of an analog signal, rather than a direct transmission of digital data.

#### Example 12-5.

The T1 transmission system requires that the input bit stream have at least a single "one" out of every eight bits and no more than 15 "zeros" in a row. This requirement is usually met by insuring that the all-zero octet of eight bits is never transmitted. When the bit



**Figure 12-7.** Power spectrum for the AMI encoder output, neglecting the effect of the transmit pulse  $g(t)$ , for different density of "ones" in the information bit stream.

stream is an encoded analog signal, the possibility of long strings of "zeros" is precluded by simply ruling out the all "zero" quantizer level (with a slight increase in quantizer error).  $\square$

In retrospect, the choice of AMI precluded direct transmission of user-provided data without an intermediate coding step to eliminate the all-zero sequence. More recent digital transmission systems have adopted one of several modifications to AMI, described in the problems, to circumvent this shortcoming.

### 12.2.2. Block Line Codes

AMI is an example of a line code that operates continuously on the input bit stream to generate a stream of data symbols. In a *block code*, the input stream is divided into blocks, each of which is translated into a block of data symbols. Assume a block of  $k$  bits is mapped into a block of  $n$  data symbols drawn from an alphabet of size  $L$ , with the constraint

$$2^k \leq L^n. \quad (12.21)$$

When equality is not met in (12.21), redundancy is available that can be used to accomplish desirable goals such as minimizing baseline wander or providing energy for timing.

#### B6ZS and HDBk

The AMI code is so commercially important that a number of codes have been invented that fix the major problem with this code — the possibility of transmitting a signal with no timing energy for an arbitrary period of time. This problem is fundamentally due to the linearity of the AMI code, since linearity implies that the all-zero bit sequence is translated into a transmitted all-zero signal. The solution to this problem is to modify the line code to make it nonlinear. This class of codes modifies AMI by performing a substitution for a block of  $k$  consecutive "0"'s that would otherwise be transmitted. The substituted block, which contains one or more non-zero symbols to ensure timing energy, uses the fact that only  $2^{k+1}$  patterns of  $k$  transmitted symbols ( $2^k$  for each of the two values of the RDS at the beginning of the block) are allowed by AMI; one of the non-allowed blocks is substituted for the all-zero block. At the receiver, this non-allowed block can be recognized and replaced with the all "zero" decoded block.

This approach complicates the coder and decoder slightly, reduces the in-service monitoring capability of the code slightly, and increases the RDS. In spite of these disadvantages, these codes are widely used because of the critical need for reliable timing recovery. Two examples of these codes are considered in the problems — B6ZS in Problem 12-12 and HDBk in Problem 12-13.

#### kBnT Codes

AMI and its derivative pseudoternary codes transmit only one bit per symbol, whereas the capacity of a ternary symbol is  $\log_2 3 = 1.58$  bits. In addition, AMI gives little control over the power spectrum. A much broader class of pseudoternary codes with the designation "kBnT" address these shortcomings, where  $k$  is again the number of information bits and  $n$  is the number of ternary symbols per block. If we choose

the largest  $k$  possible for each  $n$ , we get a table of possible codes (up to  $k = 7$ ):

$k$	$n$	Code	Efficiency
1	1	1B1T	63%
3	2	3B2T	95%
4	3	4B3T	84%
6	4	6B4T	95%
7	5	7B5T	89%

In this table, we define the *efficiency* of the code as the ratio of the *rate* of the code in bits per symbol to  $\log_2 3$ . AMI is an example of a 1B1T code. As the block size increases we can generally achieve greater efficiency, but not without cost. Greater efficiency implies better noise immunity on many channels, since it translates into a lower symbol rate for a given bit rate and hence a reduced noise bandwidth. However, greater efficiency also implies reduced redundancy, and hence less control over the statistics of the transmitted signal (power spectrum, timing recovery, density of ones, etc.). The 4B3T code seems to be a reasonable compromise between these competing goals, and has been widely studied and used in some digital subscriber loop applications [9].

A complication in kBnT line codes is the need for the decoder to know the boundaries of the blocks of  $n$  ternary symbols. This would normally be accomplished by framing, which is required in multiple channel systems in any case.

In designing kBnT codes, we must first recognize that we cannot obtain reasonable efficiency with zero-disparity code words; that is, code words with a digital sum of zero. For example, with three ternary digits, there are only six ternary code words with zero-disparity (assuming no use of the all-zero codeword, which would lead to long strings of zeros as in AMI). But as with AMI, we may define a set of different mappings (called *modes*) between binary words and ternary words, with the use of feedback to choose the appropriate mode at each block so as to minimize the RDS. Such codes are known as a *sequence-state line codes*; the first such code was described by Franaszek in 1968 [1]. The simplest such code has two modes, like AMI, although many codes have been proposed with three and four modes. A *bimode* (two-mode) 4B3T code is illustrated in Table 12-1. The remaining specification of the code is that Mode A is chosen whenever the RDS at the beginning of the block is in the range  $-3 \leq \text{RDS} \leq -1$  and Mode B is chosen when  $0 \leq \text{RDS} \leq 2$ .

Upon examination, the design principles for this code are straightforward. There are seven zero-disparity blocks of three ternary symbols; six of these are assigned to six input blocks for both modes. These can be assigned to both modes since they do not affect the RDS at the end of the block. The seventh zero-disparity ternary block, "000", is not used because it has no timing energy. The remaining  $27 - 7 = 20$  ternary code words are assigned to the remaining 10 input blocks, the positive disparity blocks to Mode A and the negative disparity blocks to Mode B. Whenever the RDS is positive, Mode B is used to make the RDS smaller, and conversely Mode A is used to make the RDS larger when it is negative. The largest change in the RDS over one block is three. The RDS at the ends of the blocks is in the range  $-3 \leq \text{RDS} \leq 2$ .

Binary Input Block	Ternary Output Block		Block Digital Sum
	Mode A	Mode B	
0000	+0-	+0-	0
0001	-+0	-+0	0
0010	0+-	0+-	0
0011	+0	+0	0
0100	++0	-0	$\pm 2$
0101	0++	0--	$\pm 2$
0110	+0+	-0-	$\pm 2$
0111	+++	---	$\pm 3$
1000	++-	--+	$\pm 1$
1001	-++	+--	$\pm 1$
1010	+-+	-+-	$\pm 1$
1011	+00	-00	$\pm 1$
1100	0+0	0-0	$\pm 1$
1101	00+	00-	$\pm 1$
1110	0+-	0+-	0
1111	-0+	-0+	0

**Table 12-1.** An example of a 4B3T code illustrating a bimode block line code.

Examination reveals that the RDS can increase by one after the first ternary symbol in the block, so that within the block the RDS is bounded by  $-4 \leq \text{RDS} \leq 3$ , and the DSV is seven. Hence, we pay a fairly substantial price in RDS for the greater efficiency of the 4B3T as compared to the 1B1T (AMI) code.

The decoder for 4B3T simply slices the ternary data, and does a table lookup to determine the binary block. Note that this is a memoryless function, independent of the state of the encoder or the RDS. Hence, there is no mechanism for error propagation. This memoryless decoder throws away all information about the sequence of states at the encoder; this additional information could be exploited by a sequence-state ML detector (the Viterbi algorithm of Chapter 8) to reduce the error rate.

### Binary Block Codes

A special case occurs when  $L = 2$  in (12.21), which means that the signal is binary. For this case, we have the simpler constraint that

$$k \leq n; \quad (12.22)$$

in other words, the block of  $n$  binary data symbols must be longer than the number  $k$  of information bits at the input to the line coder. Binary block codes are useful for media that are not well-suited to transmitting other than two-level signals, or when the additional bandwidth required for a binary transmitted signal is easier to achieve than increasing the number of levels.

#### Example 12-6.

In optical fibers (Section 5.3), intensity modulation is usually used, so that the information content is transmitted as signal intensity or power. Hence, it is possible to transmit only zero and positive levels, not negative levels. Furthermore, a modest increase in the symbol rate usually costs little in terms of faster electronics, noise immunity, or achievable distance

between repeaters. □

### Example 12-7.

In magnetic recording (Section 5.7), the medium is highly nonlinear unless a.c. bias recording is used. It makes sense to operate the medium in a binary saturation mode, with one of two magnetic polarizations. Further, it is possible to achieve a high effective information rate by encoding the information by the location of transitions in the waveform. The bandwidth of the waveform is therefore less important than the accuracy with which the location of a transition can be generated and detected. Increasing this accuracy, and hence the effective symbol rate, costs little in terms of noise immunity. □

One primary motivation for the design of line codes has been the elimination of the d.c. content of the coded signal because of a.c. coupling to the medium. It might appear that this problem does not occur for media such as optical fiber and magnetic recording, since transformers are not required. However, it is difficult to build d.c.-coupled high-speed electronics for preamplification, etc. Furthermore, the design can often be simplified by choosing as high a cutoff frequency as possible.

Binary block codes with no d.c. content can be designed by maintaining a balance between the number of positive and negative transmitted symbols in each code-word. This is usually a key objective in the design of the line code.

We can choose either a *zero-disparity code* or a *bimode code*. In a zero-disparity code [10,11], each block of  $n$  transmitted bits is constrained to have  $n/2$  "ones" and  $n/2$  zero bits, thus maintaining an RDS = 0 at the end of each block. Obviously  $n$  must be an even number. The number of possible code words is precisely

$$N = \frac{n!}{(n/2)!(n/2)!} \quad (12.23)$$

in accordance with the number of possibilities for locating  $n/2$  "ones" in  $n$  positions. The number of available code words and information capacity are tabulated in the following table:

$n$	$N$	$\log_2 N$	$k$	Efficiency
2	2	1	1	50%
4	6	2.58	2	50%
6	20	4.32	4	67%
8	70	6.13	6	75%
10	252	7.97	7	70%

The efficiency generally increases with block size, although not monotonically due to the constraint that the number of codewords be a power of two.

### Example 12-8.

With  $n = 10$ , we are tantalizingly close to eight information bits, but unfortunately we must settle for seven. The efficiency is therefore 70%. Of course, we can take advantage of this additional redundancy by choosing only the 128 out of the 252 zero-disparity codewords that have the most desirable properties. For example, we might choose codewords that

have the highest timing content, or those that maintain the smallest RDS within the code-word.  $\square$

The RDS of a zero-disparity code is generally limited to the range  $-n/2 \leq \text{RDS} \leq n/2$ , although a smaller range can be achieved with a larger redundancy and lower efficiency.

A higher efficiency can be obtained by using a bimode code. In this case we include some codewords with non-zero but small disparity. We group the blocks into two modes: a positive mode, containing all blocks with positive disparity, and a negative mode, containing all blocks with negative disparity. Blocks with zero disparity can be included in both modes. The line coder selects, for each block, the mode that will reduce the magnitude of the RDS. As in the pseudoternary block codes, the code is constructed to ensure that the decoding is memoryless, independent of the state of the coder, so that there is no error propagation mechanism.

While we have emphasized the use of line coding to ensure zero d.c. content in the signal, a code can also be used to tailor the signal to other properties of the channel.

#### Example 12-9.

A  $(d, k)$  code is often used in magnetic recording. In a  $(d, k)$  code, the coded data bits meet the constraint that the number of consecutive zeros must be at least  $d$  and at most  $k$ . The IBM 3380 disk magnetic storage system uses a  $(2, 7)$  code; i.e., runs of zeros are always at least two and at most seven in length [12]. The magnetic medium is characterized by the maximum allowable flux changes per inch (FCI) along the recording track; the binary data symbols are encoded as flux changes. With  $(d, k)$  codes with  $d > 0$ , it is possible to achieve more binary data symbols per inch than the FCI. We can write symbols at a density of  $\text{FCI} \cdot (d+1)$  binary symbols per inch without violating the condition that there be no more than FCI flux changes per inch. In effect we are transmitting faster than the FCI by encoding the information as the interval between transitions, and increasing the resolution while maintaining the FCI constraint. The upper bound  $k$  on the number of consecutive zeros is dictated by timing recovery considerations, since a zero is encoded as no flux change.  $\square$

For a  $(d, k)$  code define  $C(d, k)$  as the maximum number of information bits that can be achieved per coded binary symbol. Of course

$$C(d, k) \leq C(0, \infty) = 1. \quad (12.24)$$

Then the increase in the recording density is  $(d+1) \cdot C(d, k)$ . The calculation of the capacity is rather complicated [12,13].

### 12.2.3. Variable-Rate Codes

Some media and multiplexing methods allow the transmitted symbol rate to be variable. The philosophy is to meet the constraints of the channel and at the same time minimize the *average* transmitted symbol rate.

**Example 12-10.**

In a magnetic or optical recording system, the number of recorded symbols per information bit need not be predictable, but we would like to minimize the average total number of symbols recorded.  $\square$

**Example 12-11.**

When *statistical multiplexing* techniques are used (Chapter 18), users' messages are interleaved with stuffing information to fill out a fixed-rate bit stream. The length of a user's message is unpredictable, and after coding the number of transmitted symbols need not be predictable.  $\square$

As a simple but important practical example of a variable-rate coding scheme, consider *bit-stuffing* to meet a  $(0, k)$  run-length constraint. Here the objective is to ensure that no more than  $k$  consecutive zeros occur in a coded binary sequence (usually to meet timing recovery constraints). The technique is simple and effective — simply *add* or *stuff* an extra "one" after every  $k$  "zeros". The decoder simply removes the obligatory "one" at the end of every  $k$  consecutive "zeros".

**Example 12-12.**

A  $(0, 2)$  run-length limited binary code would encode the sequence "100011001" as "10010110011". The decoder simply replaces every sequence "001" by "00" to recover the original bit sequence. Note that the fragment "001" was mapped into "0011"; the addition of the stuffed "1" would seem to be unnecessary because the original sequence met the run-length constraints. However, the stuffed bit is necessary for correct decoding. The coded sequence is longer than the uncoded sequence (eleven symbols vs. nine bits). The length of the coded sequence is dependent on the input information bits, and hence the code has variable rate.  $\square$

The number of coded symbols can be predicted only statistically from the statistics of the information bits.

**Exercise 12-5.**

Assuming the information bits are independent and identically distributed with  $q$  the probability of a "zero", show that the coded sequence has average bit rate

$$1 + \frac{(1-q)q^k}{1-q^k} \approx 1 + (1-q)q^k \quad (12.25)$$

times the input bit rate. For example, for  $q = 1/2$  the overhead is approximately a fraction  $(1/2)^{k+1}$  which can be very small for large  $k$ . (HINT: Use the results of Problem 3-13.)  $\square$

## 12.3. FILTERING FOR SPECTRUM CONTROL

In Section 12.2.1, several linear line codes for introducing spectral nulls were described. For example, in the twinned binary code of Figure 12-3, the transmitted symbols were filtered by a transfer function  $(1 - D)$ . The result was a new sequence of pseudoternary symbols (three levels, but only one bit of information) with a null in the power spectrum at d.c. This approach can be generalized by passing the



transmitted symbols through an arbitrary filter. However, so far, no systematic design methodology has been introduced for designing such codes. This shortcoming will be rectified in this section, where the precoding technique derived in Section 10.2.4 will be generalized. The idea is very simple: spectral nulls can be introduced by putting an appropriate filter into the transmitter. That filter introduces ISI, but the ISI can be eliminated by precoding in the transmitter, at the expense of an expanded slicer in the receiver. In the case of a transmit filter with integer coefficients, we obtain an important practical technique known as *partial response*. AMI coding (Section 12.2.1) is a special case of partial response.

### 12.3.1. Adding Spectral Nulls Using Precoding

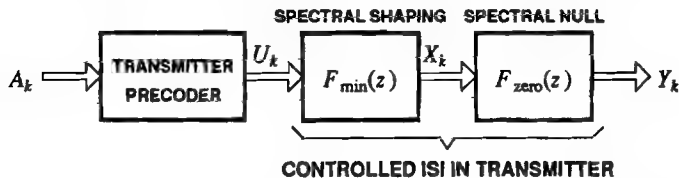
Spectral nulls can be inserted at any frequency or finite set of frequencies using a filter in the transmitter. As pointed out in Section 10.1.4, transmit filtering often has the undesirable side effect of increasing the peak transmitted power. In addition, ISI is introduced. AMI coding (Section 12.2) is an example of an approach that achieves a spectral null without ISI, at the expense of increasing the number of levels in the slicer, and a modest increase in peak and average transmitted power (for the same minimum distance).

We will now show that AMI can be generalized, using the precoding technique defined in Section 10.1.4. Recall that the purpose of precoding is to combat ISI in the transmitter, using nonlinear modulo arithmetic to avoid large increases in peak transmitted power. The precoder approach to generating spectral nulls in the transmitted signal is to put a null-generating filter in the transmitter, and then combat the resulting ISI using precoding, also in the transmitter. One of the disadvantages of precoding, the need to know the channel response accurately, is not a problem here because the filtering that introduces ISI is always accurately known to the precoder.

The basic approach, introduced in [14], is illustrated in Figure 12-8. The output of a transmitter precoder is passed through a linear filter  $F(z)$  that is monic, causal, and loosely minimum-phase. Using (2.44),  $F(z)$  can be decomposed as

$$F(z) = F_{\min}(z)F_{\text{zero}}(z), \quad (12.26)$$

where  $F_{\min}(z)$  is a monic strictly minimum-phase filter and  $F_{\text{zero}}(z)$  is a monic causal FIR filter with zeros on the unit circle. (In this subsection, we use  $z^{-1}$  in place of  $D$ ,



**Figure 12-8.** Introducing spectral nulls in the transmitter using transmitter precoding. The data symbols are  $A_k$ , and the transmitted precoded symbols are  $Y_k$ .

because this is easier to relate to the results of Section 2.6.) Since we care only about the power spectrum of the transmitted symbols  $Y_k$ , specializing  $F(z)$  to a minimum-phase filter does not limit our ability to control the power spectrum. The reason for a monic causal  $F(z)$  ( $F(\infty) = 1$ ) is that the precoder of Section 10.1.4 requires this property. Since our primary motivation is to introduce spectral nulls in the transmitted spectrum,  $F_{\text{zero}}(z)$  is the essential ingredient. A reason for choosing  $F_{\text{min}}(z) \neq 1$  will appear shortly.

If the channel is ideal, the Tomlinson precoder can be designed for the "channel" response  $F(z)$  (actually a part of the transmitter). If the channel has a non-ideal response  $H(z)$ , where  $H(z)$  is monic and causal, then the precoder can be designed for the "channel" response  $F(z)H(z)$ , which is also monic and causal. In either case, the receiver slicer must be replaced by an extended slicer.

Let  $X_k$  be the output of  $F_{\text{min}}(z)$ , as shown in Figure 12-8. An important property of  $X_k$  and  $Y_k$  is that they are bounded, since the output of the precoder  $U_k$  is bounded (because of the modulo operation in the precoder), and the filters are BIBO stable. (Because  $F_{\text{min}}(z)$  and  $F(z)$  are both minimum-phase, they have no poles on or outside the unit circle, and hence they are BIBO stable (Section 2.5.1).)

Furthermore, using the continuous approximation for the input symbols  $A_k$  (recall from Section 10.1.4 that this assumes they are i.i.d. and have a continuous uniform distribution), we saw in Section 10.1.4 that the  $U_k$  are also i.i.d. (white) and uniformly distributed. With this approximation, the power spectra of  $X_k$  and  $Y_k$  are

$$S_X(e^{j\omega T}) = \sigma_U^2 |F_{\text{min}}(e^{j\omega T})|^2, \quad S_Y(e^{j\omega T}) = \sigma_U^2 |F(e^{j\omega T})|^2, \quad (12.27)$$

where  $\sigma_U^2$  is the variance of  $U_k$ , which is approximately  $M^2/3$  according to the continuous approximation. The continuous approximation requires that the constellation be square, and  $M^2$  is the number of points in the constellation.

The motivation for including  $F_{\text{min}}(z)$  becomes apparent when we consider the case of a first-order spectral null at d.c.

### Example 12-13.

If we let  $F_{\text{zero}}(z) = 1 - z^{-1}$ , a spectral null at  $z = 1$  is introduced. An important observation is that the input  $X_k$  to the filter  $(1 - z^{-1})$  is the running digital sum (RDS) of the output  $Y_k$ . We can see this from the relation

$$\sum_{m=0}^k Y_m = X_k - X_{-1}, \quad (12.28)$$

where we can assume  $X_{-1} = 0$ . Since  $X_k$  is bounded, the RDS is bounded. Beyond this, it is advantageous to keep the RDS  $X_k$  as small as possible, for the reasons explained in Section 12.1.  $\square$

The purpose of the filter  $F_{\text{min}}(z)$  is to allow us to trade off  $\sigma_X^2$  against  $\sigma_Y^2$ . Considering the first order null of Example 12-13, it is desirable to keep  $\sigma_X^2$  small because it is a measure of the size of the RDS. We want to keep  $\sigma_Y^2$  small, because it is directly proportional to the transmitted power. As we will see, there is a direct tradeoff between these goals, in that decreasing  $\sigma_X^2$  results directly in an increase in  $\sigma_Y^2$ , and *vice versa*. Thus, there is a tradeoff between transmitted power and  $\sigma_X^2$ , and  $F_{\text{min}}(z)$

directly controls that tradeoff.

### Example 12-14.

Continuing Example 12-13,

$$\sigma_Y^2 = E[|X_k - X_{k-1}|^2] = \sigma_X^2(2 - \rho), \quad \rho = 2\operatorname{Re}\{E[X_k X_{k-1}^*]\} / \sigma_X^2. \quad (12.29)$$

Thus,  $\sigma_Y^2$  depends not only on  $\sigma_X^2$ , but also on  $\rho$ , which is related to the shape of  $S_X(e^{j\omega T})$ , which is in turn controlled by  $F_{\min}(z)$ . Thus, for a given  $\sigma_X^2$ , the transmitted power can be influenced by the choice of  $F_{\min}(z)$  through  $\rho$ . Clearly we want to choose  $F_{\min}(z)$  to minimize  $\sigma_Y^2$ .  $\square$

Since  $U_k$  is white (by the continuous approximation) and  $F_{\min}(z)$  and  $F(z)$  are both monic, it follows that  $\sigma_X^2 \geq \sigma_U^2$  and  $\sigma_Y^2 \geq \sigma_U^2$ .  $F_{\text{zero}}(z)$  is chosen in accordance with the desired location and order of the spectral nulls. The remaining design problem is then stated as follows; Minimize  $\sigma_Y^2$  subject to the constraint that  $\sigma_X^2$  is fixed. This design problem has a feasible solution as long as we choose  $\sigma_X^2 \geq \sigma_U^2$ , and as we allow  $\sigma_X^2$  to increase, the minimum  $\sigma_Y^2$  will decrease. The solution to this problem is easy at the two endpoints. Choosing  $F_{\min}(z) = 1$  allows  $\sigma_X^2 = \sigma_U^2$ , and results in the largest  $\sigma_Y^2$  we have to accept ( $\sigma_Y^2 = 2\sigma_U^2$  in the case of Example 12-13). At the other extreme, we can get an  $\sigma_Y^2$  close to  $\sigma_U^2$  by choosing  $F_{\min}(z) \approx F_{\text{zero}}^{-1}(z)$ , but doing so makes  $\sigma_X^2$  very large since  $F_{\text{zero}}^{-1}(z)$  is an unstable filter with poles on the unit circle. Thus, we can asymptotically force  $\sigma_Y^2 \rightarrow \sigma_U^2$  (its minimum), but only by allowing  $\sigma_X^2 \rightarrow \infty$ .

Following [2,14,15], the optimal  $F_{\min}(z)$  is easily found, based on the optimal linear predictor theory of Section 3.2.4. The constraint that  $\sigma_X^2$  be held constant is

$$\sigma_X^2 = \sigma_U^2 \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} |F_{\min}(e^{j\omega T})|^2 d\omega. \quad (12.30)$$

$F_{\min}(z)$  should be chosen to minimize

$$\sigma_Y^2 = \sigma_U^2 \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} |F_{\min}(e^{j\omega T})|^2 |F_{\text{zero}}(e^{j\omega T})|^2 d\omega. \quad (12.31)$$

This minimization can be solved using a Lagrange multiplier  $\lambda \geq 0$  by performing the unconstrained minimization of

$$\frac{\sigma_Y^2 + \lambda \cdot \sigma_X^2}{\sigma_U^2} = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} (|F_{\text{zero}}(e^{j\omega T})|^2 + \lambda) |F_{\min}(e^{j\omega T})|^2 d\omega \quad (12.32)$$

and then choosing  $\lambda$  to satisfy the constraint on  $\sigma_X^2$ . Minimizing (12.32) with respect to a monic minimum-phase causal  $F_{\min}(z)$  is precisely the problem of finding an optimal linear prediction error filter  $F_{\min}(z)$  for a random process with power spectrum  $(|F_{\text{zero}}(z)|^2 + \lambda)$ , which is a spectral factorization problem (Section 3.2). Writing the minimum-phase spectral factorization of this spectrum,

$$|F_{\text{zero}}(z)|^2 + \lambda = A_G^2 \cdot G(z) G^*(1/z^*), \quad (12.33)$$

where  $G(z)$  is a minimum-phase transfer function,  $\sigma_Y^2$  is minimized by choosing  $F_{\min}(z) = 1/G(z)$ . Since  $(|F_{\text{zero}}(z)|^2 + \lambda)$  is always an all-zero FIR filter,  $G(z)$  will always be an all-zero filter of the same order. Thus, the optimal  $F_{\min}(z)$  is always an all-pole filter of the same order as  $F_{\text{zero}}(z)$ , and the optimal  $F(z)$  is a filter with an equal number of zeros and poles, with the zeros on the unit circle and the poles inside the unit circle. Intuitively what the poles accomplish is to compensate for the gain introduced in  $F_{\min}(z)$  at frequencies far away from its nulls, reducing the transmitted power. Simultaneously, they have the effect of narrowing the spectral null, which in some system contexts is an undesirable side effect, and of course also increasing  $\sigma_X^2$ , which is also undesirable.

### Example 12-15.

Continuing Example 12-13, since  $F_{\text{zero}}(z)$  is first-order,  $G(z)$  will also be first-order; namely,  $G(z) = 1 - \beta z^{-1}$  for  $|\beta| < 1$ . The spectral factorization problem is thus

$$\sigma_U^2((1 - z^{-1})(1 - z) + \lambda) = A_G^2(1 - \beta z^{-1})(1 - \beta^* z). \quad (12.34)$$

Equating the coefficient of  $z^{-1}$ , we see immediately that  $\beta$  is real. With this information, we can abandon the parameter  $\lambda$ , and find  $\sigma_X^2$  and  $\sigma_Y^2$  as a function of  $\beta$ . Performing a partial fraction expansion of  $S_X(z)$ ,

$$S_X(z) = \frac{\sigma_U^2}{(1 - \beta z^{-1})(1 - \beta z)} = \frac{\sigma_U^2}{1 - \beta^2} \left[ \frac{\beta z^{-1}}{1 - \beta z^{-1}} + \frac{1}{1 - \beta z} \right]. \quad (12.35)$$

The variance  $\sigma_X^2$  is the coefficient of  $z^0$  in this expansion, which is

$$\sigma_X^2 = R_X(0) = \sigma_U^2 / (1 - \beta^2). \quad (12.36)$$

It is also helpful to know the correlation of adjacent samples,

$$R_X(1) = \beta \sigma_U^2 / (1 - \beta^2). \quad (12.37)$$

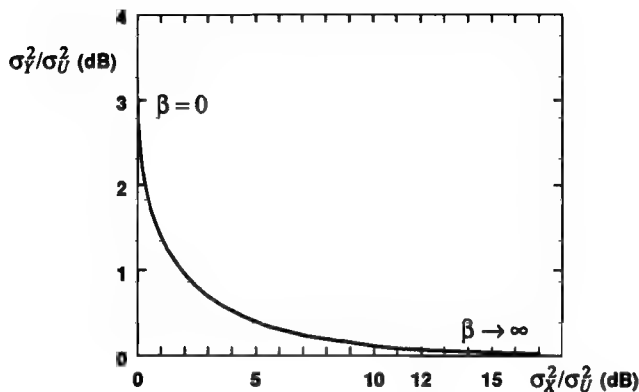
Finally,  $\sigma_Y^2$  can be determined without a need to manipulate its power spectrum,

$$\sigma_Y^2 = 2(R_X(0) - \text{Re}\{R_X(1)\}) = 2\sigma_U^2 / (1 + \beta). \quad (12.38)$$

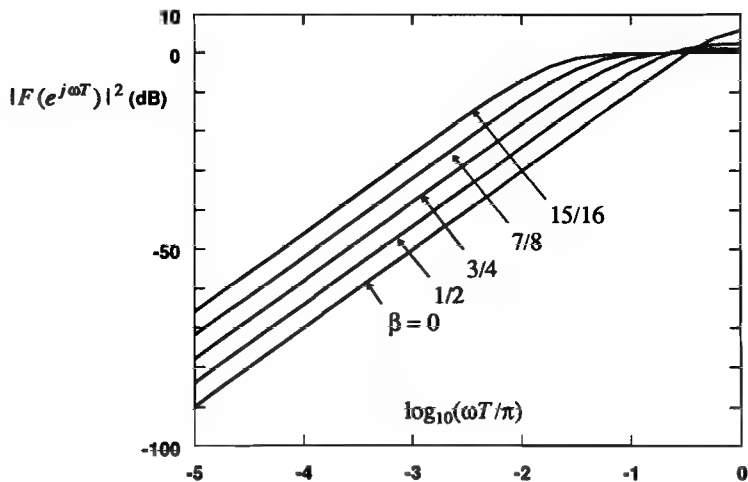
Note that  $\beta = 0$ , corresponding to no shaping filter, results in  $\sigma_Y^2 = 2\sigma_U^2$ , the worst-case transmit power. As we increase  $\beta$ , the pole location approaches the zero location, and  $\sigma_Y^2$  decreases toward  $\sigma_U^2$ , its minimum value. However, simultaneously  $\sigma_X^2 \rightarrow \infty$ , because in effect  $F_{\min}(z)$  is attempting to equalize  $F_{\text{zero}}(z)$ , which is impossible. The optimal tradeoff between  $\sigma_X^2$  and  $\sigma_Y^2$  in dB is plotted in Figure 12-9, where  $\beta$  is a free parameter that is varied to trace the tradeoff. Any point on the curve can be achieved by the optimal filter design, any point below the curve cannot be achieved by any filter design; points above the curve can be achieved by suboptimal filter designs. The magnitude response of the filter  $F(z)$  is shown in Figure 12-10. As  $\beta$  increases from zero, the width of the d.c. null decreases and the gain of the filter decreases at high frequencies (which is why the transmitted power decreases). □

## 12.3.2. Partial Response

The transmit filter  $F(z)$  just designed did not take into account implementation complexity. One of the implementation complications of the optimization procedure is that the transmitter precoder is not a finite state machine, except for very special



**Figure 12-9.** The tradeoff between  $\sigma_X^2$  and  $\sigma_Y^2$  for an optimal filter design for a first-order null at d.c.,  $F_{\text{zero}}(z) = 1 - z^{-1}$ .



**Figure 12-10.** The magnitude response of  $F(z)$  plotted against frequency on a log scale. The left side of the frequency scale approaches d.c., with the right side is  $\omega T = \pi$ , half the sampling rate. For  $\beta = 0$ , the gain at high frequencies results in a doubling of transmit power, and as  $\beta \rightarrow \infty$ , this high frequency gain is reduced and the d.c. null is narrowed.

filter designs [16], and even then the number of states in the transmitter may be very large. However, for some very simple choices of  $F(z)$ , the transmitter precoder becomes a finite-state machine with a small number of states that is simple to implement. AMI, it turns out, is an example. In particular, this happens when  $F_{\min}(z) = 1$  and  $F_{\text{zero}}(z)$  has integer-valued coefficients. The resulting system design is known as *partial response*, and has been used in many systems. The price paid for this simplicity is a larger transmitted power, relative to what can be obtained by choosing  $F_{\min} \neq 1$ , although desirably this case results in the smallest RDS.

Consider a filter response (we return to the notation  $D = z^{-1}$  used in Section 12.2)

$$F(D) = \sum_{i=0}^N f_i D^i, \quad f_0 = 1. \quad (12.39)$$

Three cases of particular interest are illustrated by the following three examples.

**Example 12-16.**

The twinned binary line code used  $F(D) = (1-D)$ , introducing a single zero at d.c. ( $D = 1$  or  $\omega = 0$ ). The discrete-time frequency response of filter  $F(D)$  is

$$F(e^{j\omega T}) = 1 - e^{j\omega T} = 2j e^{-j\frac{\omega T}{2}} \sin\left(\frac{\omega T}{2}\right), \quad (12.40)$$

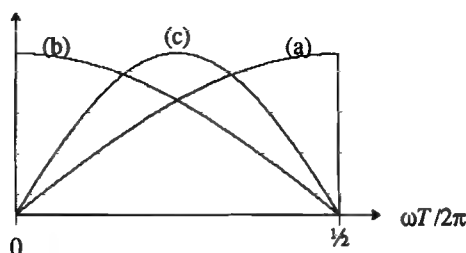
is known as *dicode*, and is plotted in Figure 12-11a.  $\square$

**Example 12-17.**

When  $F(D) = (1+D)$ , then a zero is introduced at half the symbol rate ( $D = -1$  or  $\omega = \pi/T$ ). The resulting frequency response,

$$F(e^{j\omega T}) = 1 + e^{j\omega T} = 2 e^{-j\frac{\omega T}{2}} \cos\left(\frac{\omega T}{2}\right), \quad (12.41)$$

is known as *duobinary* and is plotted in Figure 12-11b. This zero can be beneficial because it allows us to build practical digital communications systems with *no excess bandwidth*. Excess bandwidth is usually necessary because otherwise there would be a discontinuity in the spectrum of a transmitted pulse with an ideal lowpass characteristic. With a duobinary filter in the transmitter, it is possible to use zero excess bandwidth and still have no discontinuity in the spectrum.  $\square$



**Figure 12-11.** The frequency response of  $F(D)$  plotted up to half the symbol rate. (a) Dicode, (b) duobinary, and (c) modified duobinary.

**Example 12-18.**

We can achieve zeros at both d.c. and half the symbol rate by choosing

$$F(D) = (1-D)(1+D) = 1 - D^2. \quad (12.42)$$

The resulting frequency response,

$$F(e^{j\omega T}) = 1 - e^{-j2\omega T} = 2je^{j\omega T} \sin(\omega T), \quad (12.43)$$

is known as *modified duobinary* and is plotted in Figure 12-11c. This case provides the advantages of both dicode and duobinary.  $\square$

The filters in these examples place zeros in the spectrum at d.c. and/or half the symbol rate. More generally, a spectral shaping filter can be used to put an arbitrary number of zeros at these frequencies. For this case we can choose

$$F(D) = (1-D)^m (1+D)^n. \quad (12.44)$$

We allow  $m = 0$  or  $n = 0$ , but not both. Filters in this form have the special properties that the coefficients are integer-valued and the transmitter precoder has a relatively small number of states. They do not exhaust the possibilities, but include the cases of practical interest.

While filtering the data symbols with  $F(D)$  does have obvious advantages, what are the problems introduced? Two problems can be immediately recognized:

- For dicode, duobinary, and modified duobinary, if we put a binary antipodal data symbol  $\{\pm 1\}$  into the filter, the output is three-level  $\{\pm 2, 0\}$ . Since there is only one bit of information conveyed per output symbol, the resulting code is pseudoternary. Choosing other  $F(D)$ 's can result in a much larger constellation at the output (generally as  $m$  and  $n$  increase in (12.44), the size of the constellation increases). This expansion of the constellation size is the price paid for correlation of successive symbols and the resulting control over the spectrum. The larger constellation also potentially reduces noise immunity. For example, if we constrain the peak transmitted signal, the larger constellation with  $F(D)$  will result in a smaller spacing between levels.
- The filter  $F(D)$  introduces ISI. Since this has been done deliberately, rather than as a side effect of the channel, we call this *controlled ISI*.

Once we have introduced controlled ISI in order to affect the spectrum of the transmitted signal, we have several options for equalization of that ISI, as described in Chapter 10.

**Example 12-19.**

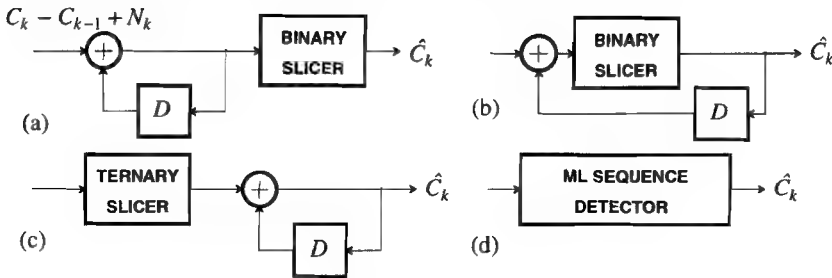
Suppose we transmit a binary antipodal signal  $C_k = \pm 1$  filtered by the dicode response  $F(D) = (1-D)$  of Example 12-16, and the equivalent discrete-time channel introduces no intersymbol interference of its own but only additive noise  $N_k$ . Four options for detecting  $C_k$  are shown in Figure 12-12. In Figure 12-12a we use a linear equalizer (LE-ZF, Chapter 10)  $F^{-1}(D) = 1/(1-D)$ , which restores a binary signal  $C_k$ ; hence a binary slicer can be used. Unfortunately, since  $1/(1-D) = 1 + D + D^2 + \dots$ , for independent noise samples the noise enhancement of this filter is infinite! More generally, a LE-ZF equalizer will suffer infinite noise enhancement for any filter of the form of (12.44) because of the algebraic zero in the frequency response. A second option, the decision-feedback equalizer of

Figure 12-12b, also uses a binary slicer, but eliminates the noise enhancement by canceling the ISI (which is postcursor) using the past decision. Unfortunately, it has error propagation. A third option exploits the ternary nature of the input signal ( $C_k - C_{k-1}$ ) by applying it directly a ternary slicer in Figure 12-12c. In the absence of noise ( $N_k = 0$ ) this slicer has no effect on the signal; hence we can place the linear equalizer filter after the slicer. This option eliminates the noise enhancement of Figure 12-12a, because the noise is removed by the slicer prior to equalization, but unfortunately also results in error propagation (Problem 12-20). The fourth option of Figure 12-12d is to use the Viterbi algorithm. This approach will have the best performance.  $\square$

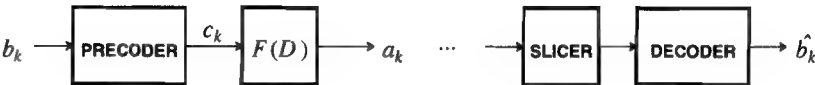
Precoding

There is another way to deal with the ISI, illustrated in Figure 12-13, which is a generalization of the approach we used to go from twinned binary to AMI in Section 12.2. We put the input bit stream through a *precoder* prior to the spectral shaping filter  $F(D)$ . In the receiver, we use a configuration similar to Figure 12-12c; namely, a slicer with more than two levels (three levels for the dicode case) followed by a *decoder* to recover the bit stream. Because of the precoder, in contrast to Figure 12-12c, the decoder is *memoryless*, and there is *no error propagation*.

The combination of a precoder with spectral shaping filter  $F(D)$  in the form (12.44) is called *partial response (PR)*. It is a simple technique for gaining the benefits of spectral shaping, with an implementation simpler than the ML sequence detector, and without the noise enhancement of the LE or the error propagation of the



**Figure 12-12.** Four options from Chapter 10 for detecting a dicode pseudoternary signal. a. A linear equalizer filter  $F^{-1}(D)$ . b. A decision-feedback filter with binary slicer. c. A filter  $F^{-1}(D)$  after a ternary slicer. d. A ML sequence detector (Viterbi algorithm.)



**Figure 12-13.** Partial response adds precoding to  $F(D)$  to allow detection without noise enhancement or error propagation.



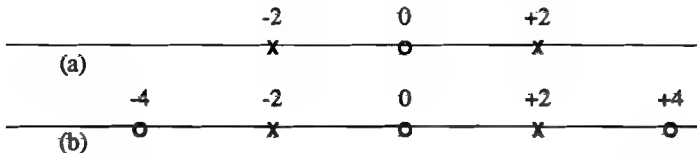
DFE. AMI is a special case of PR also known as dicode partial response. Other important examples of PR are duobinary partial response (Example 12-17) and modified duobinary partial response (Example 12-18). PR is used most often to avoid baseline wander (dicode and modified duobinary) and signal with the minimum bandwidth promised by the Nyquist criterion (duobinary and modified duobinary). Partial response is also sometimes called *correlative coding* because it introduces correlation among the transmitted data symbols, as reflected in the nulls in the spectrum. This correlation is achieved through redundancy, as in other line coding techniques. In PR, this redundancy is in the form of an increase in the size of the signal constellation over and above that required to accommodate the information bit rate.

All of the examples that we have given — dicode, duobinary, and modified duobinary PR — result in changing a binary signal into a three-level signal at the slicer. Other PR polynomials  $F(D)$  can result in a larger output constellation. It is easy to see that if a binary antipodal signal is the input to the filter  $F(D)$ , the output constellation always consists of integer values, because of the integer coefficients of the filter. In addition, the larger the order of the filter, the larger the number of integers in the output constellation.

Duobinary PR was invented by Lender [17], and was generalized in [18]. Although it is a special case of the transmitter precoding derived in Section 10.1.4, we will derive the appropriate transmitter finite-state machine by a slightly different technique. We discuss precoding next, followed by the filter design and noise considerations.

We will now limit our attention to polynomials of the form (12.44), and design the appropriate precoder. The result is a systematic method for designing precoders, and leads to the same differential precoder design for dicode PR as was considered in Section 12.2 in the context of AMI. Assume that the input to the precoder is a bit stream,  $b_k$ , and the output is binary antipodal,  $c_k = \pm 1$ . It is important to note that in (12.44),  $f_0 = F(0) = 1$ .

We have already noted that the data symbols  $a_k$  at the output of the filter  $F(D)$  have a signal constellation that includes only integers because of the integer coefficients in the filter. In fact, because of the presence of a  $(1 + D)$  or  $(1 - D)$  factor, the output constellation includes only *even* integers, as we will show shortly. The implications are as follows. Two examples of signal constellations consisting of even integers are shown in Figure 12-14, one with three points and one with five points. Recall that this constellation is redundant, and is used to communicate only one bit of



**Figure 12-14.** A three-point (a) and five-point (b) signal constellation consisting of even integers. Points in set  $\Omega_0$  are marked "o", and points in set  $\Omega_1$  are marked "x".

information. Divide this constellation into two sets of points, one set  $\Omega_0$  corresponding to input  $b_k = 0$ , and the other set  $\Omega_1$  corresponding to  $b_k = 1$ . Then the decoder can simply note whether the slicer output is in  $\Omega_0$  or  $\Omega_1$ , and put out the appropriate bit. This slicer design is memoryless, as promised.

In fact, the points in the constellation should alternate between  $\Omega_0$  and  $\Omega_1$  as shown in Figure 12-14 (where we have arbitrarily chosen  $0 \in \Omega_0$ ). Why? Since  $f_0 = 1$ , the current  $c_k$  at the precoder appears directly in the filter output. Thus, depending on  $c_k$ , the filter output is one of *two adjacent* even integers. The precoder therefore doesn't have complete control over  $a_k$ , but it can determine whether the output is in  $\Omega_0$  or  $\Omega_1$ , and thereby communicate one bit of information. At the receiver, the slicer expects a constellation of even integers, and thus applies thresholds at odd integers (two thresholds for a three-level slicer, four for a five-level slicer).

We still need to show that when the filter  $F(D)$  input is binary antipodal, the output is an even integer. Taking the case where  $F(D)$  has a factor  $(1 + D)$ , or  $n \geq 1$ , we can write

$$F(D) = J(D) \cdot (1 + D), \quad J(D) = (1 - D)^m (1 + D)^{n-1} \quad (12.45)$$

where  $J(D)$  has integer-valued coefficients. The D-transform of the filter output is

$$A(D) = F(D) \cdot C(D) = J(D)(1 + D) \cdot C(D). \quad (12.46)$$

The coefficients of  $(1 + D)C(D)$  are the sum of two coefficients of  $C(D)$  and hence must be in the set  $\{0, \pm 2\}$ . Since  $J(D)$  has integer-valued coefficients, the coefficients of  $J(D)(1 + D)C(D)$  must be even. A factor of  $(1 - D)$  rather than  $(1 + D)$  in  $F(D)$  would not change this result.

The design of the precoder is best illustrated by example.

#### Example 12-20.

*Decode partial response.* Let  $F(D) = 1 - D$ , and hence  $a_k = c_k - c_{k-1}$ . The constellation for  $a_k$  includes the three levels  $\{0, \pm 2\}$  shown in Figure 12-14a. We can fill in the following precoding table:

$c_{k-1}$	$b_k$	$c_k$	$a_k$
-1	0	-1	0
-1	1	+1	+2
+1	0	+1	0
+1	1	-1	-2

How were the entries in this table determined? For example, if  $c_{k-1} = -1$ , then  $a_k = c_k + 1$ , which assumes the values 0 or +2. By convention in Figure 12-14 we have assigned these points in the constellation to input bit 0 and 1 respectively, and this determines both  $a_k$  and  $c_k$ .

This precoding can be implemented by logic operations on the incoming data bits. If the precoder output levels  $c_{k-1}$  and  $c_k$  are represented by bits (-1 is a logic "0", and +1 a logic "1"), then the following modified table results:

$c_{k-1}$	$b_k$	$c_k$
0	0	0
0	1	1
1	0	1
1	1	0

This table can be represented as an "exclusive-or" relation  $c_k = c_{k-1} \oplus b_k$  as shown in Figure 12-15. The precoder we have derived is identical to that for AMI given in Figure 12-6. The decoder is also very simple — by convention a 0 level at the slicer is decoded as a "0", and the levels -2 and +2 are decoded as a "1".  $\square$

### Example 12-21.

*Duobinary partial response.* For this case  $F(D) = 1 + D$ , and the following table is readily developed for the precoder:

$c_{k-1}$	$b_k$	$c_k$	$a_k$
-1	0	+1	0
-1	1	-1	-2
+1	0	-1	0
+1	1	+1	+2

For duobinary, the sense is the opposite of dicode: an input "1" results in no change in the precoder output, whereas an input "0" causes a reversal. In this case error propagation without precoding results because two input sequences, 010101... and 101010... result in the all-zero sequence after filtering. With precoding, only one input sequence — all zero — results in this filter input.  $\square$

### Example 12-22.

*Modified duobinary partial response.* The PR polynomial  $F(D) = 1 - D^2$  is equivalent to interleaving two dicode PR systems with polynomial  $F(D) = 1 - D$ . Therefore we use two independent dicode PR precoders, one operating on even-numbered input bits and the other on odd-numbered input bits. This is equivalent to a two-way interleaved AMI coder, as discussed in Section 12.2.  $\square$

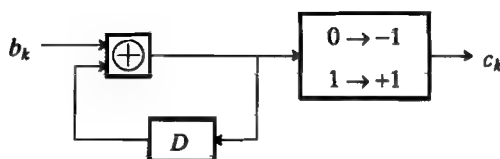


Figure 12-15. The precoder for dicode partial response.

## Partitioning of Filtering in Partial Response

We have thus far defined  $F(D)$  as the frequency response of a symbol-rate sampled filter that is a part of the transmitter. The purpose of this configuration would typically be to introduce nulls into the transmitted spectrum. In order to characterize the effect of PR on the signal spectrum, we must evaluate the effect of the precoder on the spectrum.

### Exercise 12-6.

Show that for dicode, duobinary, and modified duobinary PR, if the input bits to the precoder are independent and the probability of a "zero" is  $1/2$ , then the precoder output bits are also independent. Thus, the spectrum at the output of  $F(D)$  is affected only by the filter and not by the precoder. Note that this result is *not* valid if the input bits are not equally probable, in which case the precoder does modify the spectrum (Problem 12-22).  $\square$

We must also consider the spectrum of a continuous-time PAM signal with symbol-rate sampled response  $F(D)$ . Define an equalized pulse shape  $g(t)$  satisfying the Nyquist criterion, and then let an isolated pulse be

$$h(t) = \sum_{i=0}^N f_i g(t - iT). \quad (12.47)$$

The samples of this pulse at the symbol rate are precisely the desired PR response,

$$h(kT) = \begin{cases} f_k, & 0 \leq k \leq N \\ 0, & \text{otherwise} \end{cases}, \quad (12.48)$$

and the Fourier transform of this pulse is

$$H(j\omega) = F(e^{j\omega T})G(j\omega), \quad (12.49)$$

where  $F(e^{j\omega T})$  is plotted in Figure 12-11 for three cases. In the case of dicode PR, which is the same as AMI, if we attempted to use 0% excess bandwidth there would be a discontinuity in the spectrum at half the symbol rate; therefore, we require some excess bandwidth for this case ( $g(t)$  must have bandwidth greater than  $\pi/T$ ). For both duobinary and modified duobinary, however, it is practical to use 0% excess bandwidth because the spectrum will have no discontinuity. This is the origin of the term "duobinary"; a symbol rate double that of binary signaling is possible for a given bandwidth (actually, since 100% excess bandwidth is not required for binary antipodal signaling, the advantage is smaller than this).

With excess bandwidth,  $g(t)$  is not unique and thus the PR isolated pulse is not unique. However, for 0% excess bandwidth,  $G(j\omega)$  is uniquely an ideal lowpass filter, and the PR responses are

$$\text{Duobinary: } h(t) = \text{sinc}\left(\frac{\pi}{T}t\right) + \text{sinc}\left(\frac{\pi}{T}t - \pi\right) \quad (12.50)$$

$$\text{Modified duobinary: } h(t) = \text{sinc}\left(\frac{\pi}{T}t\right) - \text{sinc}\left(\frac{\pi}{T}t - 2\pi\right).$$

These pulses are plotted in Figure 12-16. The cancellation of the tails of the two

"sinc" functions in (12.50) results in a well-behaved response in spite of zero excess bandwidth. Also note the width of the pulses, which obviously reflect their narrower bandwidth compared to Nyquist pulses (Chapter 6).

While we have emphasized the application of PR to controlling the transmitted spectrum, which results in putting the  $F(D)$  response into the transmitter, there are other possible motivations for using PR, and reasons to make all or a portion of the filter  $F(D)$  a part of the channel or the receiver. For example, the channel may naturally place a zero in the spectrum (for example, an a.c.-coupled channel will have a zero at d.c.), in which case a part of the  $F(D)$  response may arise naturally. Alternatively, a zero placed in the receiver may help to reduce the noise at the slicer input, since it reduces the gain in the receive equalizer at some frequencies. In either of the latter cases, we can think of PR not as a method of spectrum control, but rather as an alternative to the DFE for improving noise immunity, but without the error propagation of the DFE.

Noise is an important consideration in the partitioning of filtering. It is often stated in the literature that there is a noise immunity penalty with PR. This is true on channels with little or no intersymbol interference, but is by no means generally true for other channels. We illustrate this by a simple calculation for two simple discrete-time channels. We will express the error probability in terms of the peak transmitted signal energy per symbol ( $E_{\text{PEAK}}$ ) and the average transmitted signal energy per symbol ( $E_{\text{AVG}}$ ).

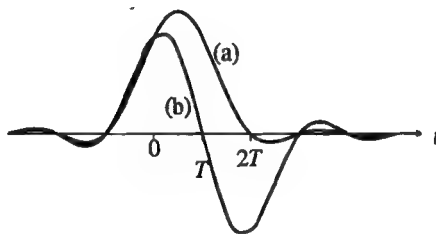
#### Example 12-23.

*No intersymbol interference.* Let a channel be represented by the model

$$Y_k = X_k + N_k \quad (12.51)$$

where the zero-mean noise samples are independent with variance  $\sigma^2$ . Consider now four strategies:

*Binary antipodal signaling with two-level slicer:* For this case we let  $X_k = \pm 1$  be a binary antipodal data symbol. The slicer input is  $\pm 1$ , and a noise sample must be larger than unity to cause an error. The error probability is therefore  $Q(1/\sigma)$ . The peak and average energies



**Figure 12-16.** Time response of equalized pulses for duobinary (a) and modified duobinary (b) with 0% excess bandwidth.

are the same,  $E = 1$ , and hence the error probability is  $Q(\sqrt{E}/\sigma)$ .

*Duobinary with transmit filtering to  $(1 + D)$ :* For this case we let  $X_k = C_k + C_{k-1}$ , where  $C_k$  is a duobinary precoded antipodal symbol, and the three-level slicer input  $Y_k = 0, \pm 2$ . Since the distance between slicer levels is two, the noise must again be larger than unity to cause an error, and the error probability will be a constant times  $Q(1/\sigma)$ . (The constant depends on end effects.) The peak transmitted energy (4) and average transmitted energy (2) are not the same (assuming independent input bits equally likely to be zero or one), so the error probability is  $Q(\sqrt{0.25E_{\text{PEAK}}}/\sigma)$  or  $Q(\sqrt{0.5E_{\text{AVG}}}/\sigma)$ . Duobinary is 3 dB poorer than binary antipodal with respect to average signal power, and 6 dB for peak signal power.

*Duobinary with receive filtering to  $(1 + D)$ .* For this case the transmitter is defined as  $X_k = C_k$ , the precoder output without the  $(1 + D)$  filter, and the receiver applies  $Y_k + Y_{k-1}$  to the three-level slicer. The variance of the noise at the slicer input is (because it is the sum of two independent noise samples) equal to  $2\sigma^2$ , and a noise larger than unity is required to cause an error. The peak and average energies are both unity, so the error probability is a constant times  $Q(1/\sqrt{2}\sigma) = Q(\sqrt{0.5E}/\sigma)$ , 3 dB worse than binary antipodal. This is because we have the same spacing of slicer levels, but twice the noise variance.

*Duobinary with transmit filtering and ML sequence detection.* For the transmit filtering case, the slicer input noise is white and we can apply ML sequence detection in place of simple slicing. The channel consists of binary antipodal symbols  $C_k$  input to channel response  $(1 + D)$ , and as shown in Chapter 8, the minimum distance corresponds to a single error. In this case the error has magnitude two, so the minimum distance is  $d_{\min}^2 = 8$ . There are an infinite number of minimum-distance error events, each corresponding to alternating precoded symbols. After decoding, each of these error events results in only a single bit error. The error probability is thus upper bounded by  $Q(\sqrt{8}/2\sigma)$ , and since the peak and average energies are again 4 and 2 respectively, the error probability is  $Q(\sqrt{0.5E_{\text{PEAK}}}/\sigma)$  or  $Q(\sqrt{E_{\text{AVG}}}/\sigma)$ . The performance is 3 dB better than for a simple slicer, and with respect to average transmitted energy is equal to binary antipodal.  $\square$

This example illustrates why it is often stated that duobinary PR has a three dB noise penalty, as this is true on channels which lack intersymbol interference. In addition, it illustrates how the ML sequence detector can gain back all but a tiny bit of the penalty. (It is not strictly equivalent because we have ignored distances larger than the minimum distance.) The reason the ML detector has an advantage is that not all sequences of pseudoternary levels at the slicer are allowed; the ML detector uses this additional information to advantage. The example also shows how precoding is beneficial when the ML detector is used — it reduces the number of errors caused by the infinite set of minimum-distance error events.

Since PR is equivalent to a DFE with feedback filter  $(F(D) - 1)$ , but without error propagation, we know from Chapter 10 that on many channels with intersymbol interference a properly chosen PR can have a noise *advantage* over binary antipodal signaling, as opposed to the disadvantage displayed in the previous example. Consider the following simple case of a channel for which duobinary is advantageous.

#### Example 12-24.

*Channel with duobinary ISI.* Let the channel of Example 12-23 be modified to

$$Y_k = X_k + X_{k-1} + N_k, \quad (12.52)$$

so the channel itself has the  $(1 + D)$  duobinary response. For this case, in order to use binary antipodal signaling, we must first equalize the channel to eliminate the intersymbol interference in either the transmitter or receiver. Let us consider both transmitter and receiver equalization, along with duobinary with slicer and ML sequence detection.

*Binary antipodal with transmitter equalization:* Here we implement a filter  $1/(1 + D)$  in the transmitter. Unfortunately, the peak transmitted signal is unbounded, since

$$\frac{1}{1 + D} = 1 - D + D^2 - \dots \quad (12.53)$$

and the transmitter output is a sum of an infinite number of binary antipodal symbols. Thus both the peak and average transmitted power are infinite.

*Binary antipodal with receiver equalization:* The LE-ZF equalizer puts a filter  $1/(1 + D)$  in the receiver. Unfortunately the SNR is zero, since the noise at the slicer input

$$N_k - N_{k-1} + N_{k-2} - \dots \quad (12.54)$$

now has infinite variance. The channel response with an algebraic zero cannot be equalized in the zero-forcing sense.

*Duobinary with no equalization:* Since the channel response is what we would like for duobinary, we can simply transmit  $X_k = C_k$  where  $C_k$  is the duobinary precoded signal, and apply the resulting channel output  $Y_k$  directly to a ternary slicer. A noise larger than unity will cause an error, so the error probability is a constant times  $Q(1/\sigma)$ . Since both the peak and average transmitted powers are unity, the error probability becomes  $Q(\sqrt{E}/\sigma)$ .

*Duobinary precoding with ML sequence detection.* Again the minimum distance is  $d_{\min}^2 = 8$ , and hence the error probability is  $Q(\sqrt{8}/2\sigma) = Q(\sqrt{2E}/\sigma)$ . There is again a 3 dB advantage over the simple slicer.  $\square$

This example illustrates the extreme case where the channel itself has a zero at half the symbol rate. We cannot use conventional Nyquist-pulse equalization since if we do the equalization in the transmitter, the peak signal becomes infinite, or if we do it in the receiver the noise becomes infinite. We must use a technique such as duobinary, or almost equivalently the DFE, and in the process we gain a large (infinite!) noise advantage. This example is extreme, since a channel with infinite loss at half the symbol rate is rare. However, channels with very large losses are common.

#### Example 12-25.

Wire-pair and coaxial cable systems (Section 5.2) often operate at their maximum ranges with losses of 60 to 80 dB at half the symbol rate. For these channels, duobinary has a substantial noise advantage over binary antipodal signaling [19]. In fact, duobinary is used in wire-pair systems to achieve a doubling of bit rate (3 Mb/s vs. 1.5 Mb/s) over AMI (dicode PR) in the T1C transmission system.  $\square$

Finally, we should reiterate that with PR precoding, although the number of levels is increased, not all sequences of data symbols are allowed (this is the redundancy). This fact can be used for performance monitoring (Problem 12-24) and, as with any channel with intersymbol interference, the Viterbi algorithm can be used to advantage in the decoding. Finally, PR can be generalized to an arbitrary number of transmitted levels (Problem 12-25).

## 12.4. CONTINUOUS-PHASE MODULATION

*Continuous-phase modulation* (CPM) is a class of signaling schemes that maintains a constant envelope and avoids abrupt phase changes. The constant envelope is advantageous in many situations, particularly for channels with nonlinearities. Phase-shift keying with a rectangular pulse shape also maintains a constant envelope, but has phase discontinuities that result in a larger bandwidth for the transmitted signal.

A CPM signal is a phase modulated carrier,

$$X(t) = K \cos \left[ \omega_c t + 2\pi h \int_{-\infty}^t S(\tau) d\tau + \phi \right], \quad (12.55)$$

where  $h$  is called the *modulation index* and

$$S(t) = \sum_{m=-\infty}^{\infty} A_m g(t - mT). \quad (12.56)$$

To maintain phase continuity,  $S(t)$  must not have impulses.

### Example 12-26.

In Chapter 6 we considered special cases of CPM where the pulse  $g(t)$  is rectangular, or constant over the interval 0 to  $T$  and zero elsewhere. In this case, CPM is called *continuous phase FSK* (CPFSK). An interesting special case of CPFSK is MSK, as described in Chapter 6. The MSK signal of Chapter 6 signal can be written

$$X(t) = \cos \left[ \omega_c t + \pi \int_{-\infty}^t S(\tau) d\tau + \frac{\pi}{2} \right], \quad (12.57)$$

where  $S(t)$  is given by (12.56),  $A_m = \pm 1$ , and

$$g(t) = \frac{1}{2T} w(t) = \begin{cases} 1/2T; & 0 \leq t < T \\ 0; & \text{otherwise} \end{cases}. \quad (12.58)$$

The modulation index is  $h = 1/2$ .  $\square$

### Exercise 12-7.

Show that (12.57) can be written in the form (6.152). Identify  $b_m$  and  $\phi_m$  in (6.152) and show that they satisfy (6.153).  $\square$

It is common to normalize the pulse  $g(t)$  (as in (12.58)) so that it integrates to  $1/2$ . With this normalization, the CPM signal has a phase change of  $A_m h \pi$  radians in one symbol interval, with respect to the carrier  $\omega_c$ .

MSK provides a constant envelope signal with considerably narrower bandwidth than a constant envelope PSK (using rectangular pulses). However, since for CPFSK  $g(t)$  in (12.56) is rectangular, there is a discontinuity in the first derivative of the CP



signal. This implies that with smoother choices for  $g(t)$  we can significantly reduce the bandwidth by ensuring continuous first, or even second or third, derivatives. The simplest case, called *full-response CPM*, uses a  $g(t)$  that is zero outside the interval  $0 \leq t \leq T$ . The second case, called *partial-response CPM*, uses a  $g(t)$  that extends over several symbol intervals. The term partial response, as in Section 12.3, refers to the deliberate introduction of ISI for spectrum control. In fact, the spectral properties of partial-response CPM signals are considerably better than full-response CPM and CPFSK [20].

The evolution of a CPM signal over time can be compactly described using a *phase diagram*. The phase diagram plots the phase term from (12.55)

$$2\pi h \int_{-\infty}^t S(\tau) d\tau \quad (12.59)$$

for all possible input symbols  $A_m$ .

#### Example 12-27.

The phase diagram for MSK is shown in Figure 12-17.  $\square$

Since phase is modulo  $2\pi$ , the phase diagram is best viewed wrapped around a cylinder with circumference  $2\pi$ . If the modulation index  $h$  is rational, then the phase diagram will have a finite number of points on this cylinder. In this case, it can be viewed as the trellis diagram for a Markov chain. Since the phase evolves according to a Markov chain, the Viterbi algorithm is commonly used in the detector to perform optimal sequence detection.

#### Example 12-28.

In Chapter 6 we found a receiver structure (Figure 6-46) for MSK signals that performs roughly as well as the optimal matched-filter receiver for orthogonal FSK signals. However, by taking the absolute value of the sampled output of the matched filters, that structure discards useful information. The useful information that is discarded is easily seen in

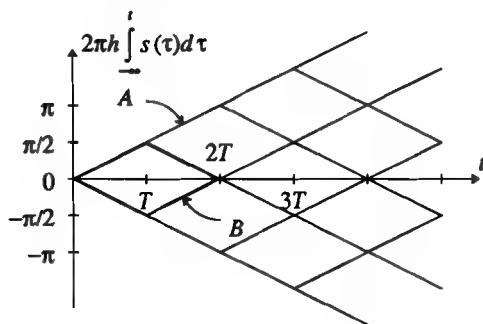


Figure 12-17. A phase diagram for an MSK signal.

Figure 12-17. The receiver in Figure 6-46 makes no distinction between the path labeled "A" and the path labeled "B", which correspond to signals that are  $\pi$  radians apart, or antipodal. Obviously, making the distinction would be useful. Considering that the phase is modulo  $2\pi$ , the phase diagram in Figure 12-17 has only four states, best viewed as lying on a cylinder. The minimum-distance error event has length two and is the bold diamond shape in Figure 12-17 (see Problem 12-26). Furthermore, the squared distance of this error event is twice the squared distance between the two orthogonal signaling pulses, so application of the Viterbi algorithm results in a 3 dB improvement over the receiver in Figure 6-46. That receiver was shown to perform roughly as well as an optimal orthogonal FSK receiver, which is 3 dB worse than an optimal binary antipodal (2-PSK) signal of the same average power. Consequently, by using the Viterbi algorithm with MSK we are able to recover the 3 dB loss associated with FSK and match the performance of PSK.  $\square$

Construction of the trellis and application of the Viterbi algorithm becomes more complicated for CPM signals other than MSK. The number of states depends on the modulation index  $h$  and the extent of the pulse  $g(t)$ .

## 12.5. SCRAMBLING

Scrambling is a method of achieving d.c. balance and eliminating long sequences of zeros to ensure accurate timing recovery without redundant line coding. Scramblers use *maximum-length shift registers (MLSR)* on the input bit stream to "randomize" or "whiten" the statistics of the data, making it look more random.

Practical data transmission systems have no control over the bit sequences which the user is going to transmit. There are particular bit sequences, such as long strings of zeros or ones, which occur very often in practice and which can cause difficulties. At the theoretical level, these sequences strongly violate the assumption that the input sequence is random and i.i.d. On a more practical level, they can cause problems such as excessive radio frequency interference (RFI), crosstalk, and difficulty in timing recovery and adaptive equalization.

Any technique without redundancy such as scrambling must perform a one-to-one mapping between input data bit sequences and coded bit sequences. The objective is to map sequences that are problematic and fairly likely to occur (such as all zeros) into a coded sequence which looks more random and is less problematic. However, since the mapping is one-to-one, there must also be an input sequence that maps into a problematic sequence, such as all zeros! We just hope that this input sequence is very improbable. Thus in general, redundant line coding is a safer method of achieving our desired objectives, but scrambling is attractive and often used on channels with extreme bandwidth constraints because it requires no redundancy.

### Example 12-29.

All CCITT-standardized voiceband data modems incorporate scrambling. This is attractive because of the desire to maximize spectral efficiency.  $\square$

If a binary antipodal signal is wide-sense stationary, then the power spectral density after scrambling this signal is essentially flat down to d.c., although there is no

discrete component at d.c. even if the original user bits have such a component (with some exceptions to be described later). This implies that an a.c. coupled medium is permissible, although the cutoff frequency has to be quite low to avoid appreciable baseline wander. Alternatively, some other line coding scheme, such as AMI, can be added to insert a rational zero at d.c. A combination of AMI and scrambling would be effective in eliminating low frequency components as well as insuring adequate timing energy. Often scrambling alone is combined with *quantized feedback*[21] to compensate for the baseline wander. Quantized feedback is a form of decision-feedback equalization (Chapter 10) specifically designed to compensate for the baseline wander ISI using past decisions.

There are two forms of scrambling — *self-synchronizing* and *frame-synchronized*. Both types of scramblers use *maximal-length shift-register sequences*, which are periodic bit sequences with properties that make them appear to be random. These sequences are also called *pseudorandom sequences* because of their apparent randomness. A pseudorandom sequence generated by an  $n$ -bit shift-register is a binary sequence with period  $r = 2^n - 1$ .

Pseudorandom sequences are generated by a *feedback shift register* as pictured in Figure 12-18. This device is governed by the relation

$$x_k = h_1 \cdot x_{k-1} \oplus \cdots \oplus h_n \cdot x_{k-n} \quad (12.60)$$

where the summation is modulo-two, the output  $x_k$  is binary assuming the values "0" and "1", and similarly the coefficients of the shift register are binary. The zero coefficients correspond to no feedback tap, whereas the one coefficients correspond to the direct connection of the shift register output to the modulo-two summation.

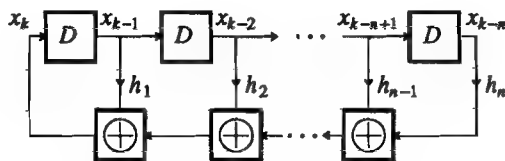
### Example 12-30.

A simple case is  $n = 2$  and  $h_1 = h_2 = 1$ . For this case,

$$x_k = x_{k-1} \oplus x_{k-2}. \quad (12.61)$$

□

Adding  $x_k$  to both sides of (12.60), and recalling that  $x_k \oplus x_k = 0$ , we get



**Figure 12-18.** A linear feedback shift register with binary input. The coefficients are binary, and the summation is modulo-two.

$$x_k \oplus h_1 x_{k-1} \oplus \cdots \oplus h_n x_{k-n} = 0. \quad (12.62)$$

In other words,

$$x_k * h_k = 0 \quad (12.63)$$

if we define  $h_0 = 1$  and  $h_m = 0$  for  $m < 0$  and  $m > n$  and of course we interpret the summation in the convolution in the modulo-two sense. The D-transform of (12.63) is

$$h(D)X(D) = 0 \quad (12.64)$$

where

$$h(D) = 1 \oplus h_1 D \oplus \cdots \oplus h_n D^n \quad (12.65)$$

is the transfer function of the shift register. (12.65) is called a *modulo-two D-transform*, and is used extensively in Chapters 13 and 14. Given any binary sequence  $b_k$  (deterministic or random), the modulo-two D-transform is

$$B(D) = \cdots \oplus b_{-1} D^{-1} \oplus b_0 \oplus b_1 D \oplus b_2 D^2 \oplus \cdots \quad (12.66)$$

where  $\oplus$  denotes modulo-two addition. In other words, it is just like a Z-transform, except that the additions are modulo-two and the symbol  $D$  is used instead of  $z^{-1}$ . Now any convolution of two sequences

$$C_k = g_k * B_k \quad (12.67)$$

can be written in the "D-domain" as

$$C(D) = G(D)B(D). \quad (12.68)$$

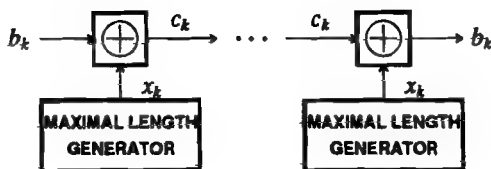
The transfer function  $h(D)$  for the generator is a polynomial of degree  $n$  (we assume that  $h_n = 1$ ) with binary coefficients, and is given the special name *generator polynomial*. There is a one-to-one correspondence between generator polynomials and feedback shift registers. Further mathematical properties of shift-register generators are discussed at some length in Appendix 12-A. Shift-register generators are also widely used as spreading sequences in spread spectrum (Section 8.6), because pseudo-random sequences have a constant-magnitude DFT (with the exception of the zero-frequency coefficient), making them suitable for generating broadband pulses with a narrow autocorrelation function.

### 12.5.1. Frame-Synchronized Scrambler

A frame-synchronized scrambler [22,23], also called a *cryptographic scrambler*, pictured in Figure 12-19, performs a modulo-two summation of the user's bit stream  $b_k$  with the output  $x_k$  of a maximal-length feedback shift-register in the transmitter to generate the scrambled bit stream  $c_k$ ,

$$c_k = b_k \oplus x_k. \quad (12.69)$$

The scrambled bit stream is transmitted to the receiver by whatever line coding method is chosen, where the stream is descrambled by another modulo-two summation with the output of another identical generator to recover the original user's bit stream. This recovery follows from the relation



**Figure 12-19.** A frame-synchronized scrambler where the maximal-length generator is as in Figure 12-18.

$$c_k \oplus x_k = b_k \oplus x_k \oplus x_k = b_k \quad (12.70)$$

since  $x_k \oplus x_k = 0$ .

With binary antipodal signaling, if there were no scrambling then  $b_k = 1$  would be transmitted as  $a_k = +1$  and  $b_k = 0$  would be transmitted as  $a_k = -1$ . Similarly, we can define a binary antipodal version of the generator sequence  $x_k$ , call it  $s_k$ , that assumes the values  $\pm 1$ . With scrambling, we substitute  $c_k$  for  $b_k$ . We can easily see that this is equivalent to *multiplying* the binary antipodal version of the user's bits by the negative of the binary antipodal maximal-length sequence  $s_k$ ; that is, we transmit  $-s_k a_k$  in place of  $a_k$ .

The sequence  $s_k$  is periodic, and hence consists of a sequence of equally spaced harmonics, where the spacing of harmonics is the sampling rate divided by the period  $r$ . In fact, it is shown in Appendix 12-A, Exercise 12-14, that the magnitude of all the harmonics, with the single exception of the d.c. harmonic, are equal ((12.90) and (12.94)). Thus, the scrambler is equivalent to modulating by a set of equal-amplitude harmonics equally-spaced across the band and summing the results. We would expect the result to be approximately white almost without regard to the spectrum of the original user bit stream. In quantifying this concept, we have to deal as always with the cyclostationary nature of a modulated signal.

#### Exercise 12-8.

Assume the binary antipodal version  $A_k$  of the user bit stream  $B_k$  is a wide-sense stationary random process with autocorrelation function  $R_a(l)$ .

- Show that the scrambled sequence is cyclostationary.
- Average the autocorrelation over one period, and show that as  $r \rightarrow \infty$  the power spectrum approaches  $R_a(0)$ . Hence, as  $r$  gets large, the spectrum becomes white independent of the spectrum of the user's bit stream.  $\square$

The correct operation of the frame-synchronized scrambler depends on the alignment in time of the two maximal-length sequences of period  $r = 2^n - 1$  in the scrambler and descrambler. This is called *frame-synchronization*, and is accomplished by an additional frame synchronization mechanism.

If the user should provide the scrambler with the maximal-length sequence itself, the scrambled sequence will be all zeros! However, this eventuality should be very

improbable, unless the user is deliberately attempting to sabotage his own transmission. More generally, the periodic structure of the generator output implies that the scrambled sequence will be periodic whenever the input stream is periodic.

**Exercise 12-9.**

Show that when the input stream  $b_k$  has period  $s$  and the generator output has period  $r = 2^n - 1$ , then the scrambler output will be periodic with period equal to the least common multiple (LCM) of  $s$  and  $r$ . (This does not preclude the period being a divisor of this LCM, as we will see below.) In particular, when  $r$  is prime, which we can arrange, then this LCM period is  $sr$ , a multiple of the period of the maximal-length sequence.  $\square$

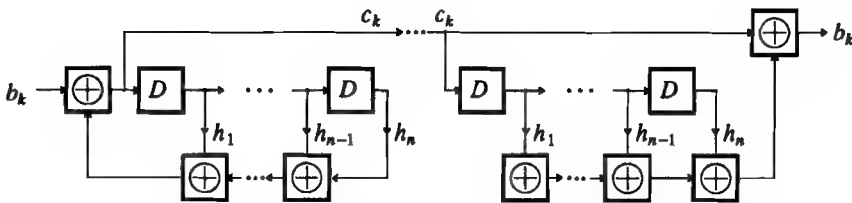
We will generally maximize the period of the output, which is desirable, by choosing  $r$  to be a prime number, such as  $r = 2, 3, 7, 31, \dots$ . For this case, the smallest LCM will occur when  $s = 1$ , or in other words the input is all zeros or all ones.

A serious problem occurs when the input stream has period  $r$  or some multiple of  $r$ , since the scrambled stream can then have a much shorter period than the LCM and can also have a large d.c. component resulting in severe baseline wander. If  $r$  is large this pathological situation will arise with vanishingly low probability, and need not be of great concern.

### 12.5.2. Self-Synchronized Scrambler

We can avoid the necessity for frame synchronization of the scrambler by using the self-synchronized scrambler [22,23] of Figure 12-20. For this case, we use a shift-register generator in the transmitter, except that we add the input stream directly to the input of the shift-register. The shift-register input  $c_k$  is also the scrambled stream, and is applied to the input of an identical shift-register in the descrambler. Since both shift-registers, the one in the scrambler and the one in the descrambler, have the same inputs (in the absence of transmission errors), and the shift-register output is added modulo-two in the scrambler and descrambler, it follows that the input stream  $b_k$  is recovered by the descrambler.

Mathematically, the scrambler is represented by the relation



**Figure 12-20.** A self-synchronized scrambler.

$$c_k = b_k \oplus h_1 c_{k-1} \oplus \cdots \oplus h_n c_{k-n} \quad (12.71)$$

and taking the D-transform of both sides we get

$$h(D)C(D) = B(D) \quad (12.72)$$

where  $h(D)$  is the same generator polynomial as in the maximal-length generator. Formally, we can write

$$C(D) = \frac{B(D)}{h(D)}, \quad (12.73)$$

and we can view the scrambler as dividing the polynomial corresponding to the input stream by the generator polynomial  $h(D)$ , whereas the descrambler multiplies the scrambled stream polynomial by  $h(D)$ , from (12.72).

#### Example 12-31.

The CCITT V.22bis voiceband data modem uses a self-synchronizing scrambler with generating polynomial

$$h(D) = 1 \oplus D^{14} \oplus D^{17}. \quad (12.74)$$

The V.26ter modem uses two polynomials,

$$h(D) = 1 \oplus D^{18} \oplus D^{23} \quad (12.75)$$

in one direction of transmission and

$$h(D) = 1 \oplus D^5 \oplus D^{23} \quad (12.76)$$

in the other direction. The reason for using two generators is that the V.26ter modem uses echo cancellation to separate the two directions of transmission (Chapter 19), and it turns out to be important to ensure that the scrambled streams in the two directions are uncorrelated.  $\square$

Because of the linearity of the scrambler circuit, and in spite of the modulo-two arithmetic, we can view the processing as follows. The scrambler consists of an all-pole filter with transfer function  $1/h(D)$ ; the descrambler consists of an all-zero filter with transfer function  $h(D)$ . The product of the two transfer functions is unity, recovering the original bit stream. The all-pole filter output can be decomposed into the superposition (modulo-two summation in this case) of two solutions — the zero-input solution (transient response) and the zero-state solution (steady-state response). The zero-input solution is precisely the maximal-length sequence used in the frame-synchronized scrambler, where this solution persists forever and does not die down as it might in a normal filter. In this view of the scrambler, the output is the all-pole filtered version of the input stream added to the maximal-length sequence. The latter gives us the "randomization" operation which we are attempting to achieve with the scrambler.

The self-synchronized scrambler works without any alignment of the scrambled sequence for reasons stated earlier. However, it does have one disadvantage — error propagation. When the input to the descrambler shift register is different from that of the scrambler shift register due to a transmission error, this causes additional errors. Specifically, there is one direct error and one secondary error for each non-zero tap in the shift register. The error multiplication is therefore by a factor equal to one plus the number of non-zero taps.

The self-synchronizing scrambler also has more problems with periodic input streams than the frame-synchronized scrambler[22]. Specifically, when the input has period  $p$ , the output will have one of the following periods depending on the initial state. For one particular state (with probability  $2^{-n}$ ), the period will be  $p$ , which could be very unfortunate when  $p$  is very small. For the remaining states (with probability  $1 - 2^{-n}$ ), the output has period equal to the LCM of  $p$  and  $r$ , the same as the frame-synchronized scrambler.

### Example 12-32.

Using the second-order generator polynomial of Example 12-30, the scrambler is given by

$$c_k = b_k \oplus c_{k-1} \oplus c_{k-2}. \quad (12.77)$$

Assume that we start a periodic input just at the point that the state happens to be  $(c_{k-1}, c_{k-2}) = (0, 0)$ . Then we can see that the response to the all-zero input from that point is all-zero (the same period,  $p = 1$ , as the input). For any other initial state the output has period  $1 \cdot 3 = 3$ . As another example, assume the input is alternating zero-one (period two). Then for an initial state  $(0, 1)$  the output is also alternating zero-one (period two), whereas for any other state the output has period  $2 \cdot 3 = 6$ .  $\square$

The probability of a short periodic output can be minimized by choosing  $n$  large.

## 12.6. FURTHER READING

Line coding is a practical subject that is not widely covered in textbooks. One excellent reference is the tutorial article by Duc and Smith [4]. There is some coverage in a recent digital communications textbook [24]. Recent results in combining line coding with trellis coding are reviewed in [2], and recent results in line codes for magnetic recording are discussed in [25].

Partial response is covered in a tutorial paper [26] and book chapter [27]. The performance of the Viterbi algorithm for partial response systems was determined by Kobayashi [28] and Forney [29].

A general treatment of full-response continuous-phase modulation is given by Aulin and Sundberg [30], and of partial response CPM by Aulin, Rydbeck, and Sundberg. A general discussion is also given by Simon [31]. For a general approach to modeling the phase evolution as a Markov chain, see [32]. The same issue of the IEEE Transactions on Communications (March 1981) has several useful papers on CPM in a special section. For tutorials on MSK, we recommend Pasupathy [33] and Haykin [34].

Finite fields and maximal-length shift register sequences are covered in detail in [35].



## APPENDIX 12-A

### MAXIMAL-LENGTH FEEDBACK SHIFT REGISTERS

In this appendix we will consider the properties of a periodic sequence generated by the shift register circuit of Figure 12-18 with generator polynomial  $h(D)$ . While a full treatment of this problem requires some sophisticated mathematics, we can understand most of the properties of this generator using only elementary concepts.

Mathematically, the binary coefficients of the generator polynomial together with the rules of multiplication and modulo-two addition constitute an *algebraic field*, similar to the real or complex numbers. In recognition that this field has only two elements, it is also called a *finite field* or *Galois field with two elements*  $GF(2)$ . In general there exists a single finite field with a number of elements equal to any prime integer to any power. We will limit ourselves here to finite fields with two elements, which is just the modulo-two arithmetic considered in this chapter. The more general case is discussed in Appendix 13-A.  $GF(2)$  has two elements "0" and "1", and modulo-two arithmetic is used.

#### Example 12-33.

As an illustration of polynomial arithmetic over  $GF(2)$ , multiplying the polynomials  $(1 \oplus D)$  and  $(1 \oplus D \oplus D^2)$ ,

$$(1 \oplus D)(1 \oplus D \oplus D^2) = 1 \oplus D \oplus D \oplus D^2 \oplus D^2 \oplus D^3 = 1 \oplus D^3. \quad (12.78)$$

We have used the fact that, for example,

$$D \oplus D = (1 \oplus 1)D = 0 \cdot D = 0. \quad (12.79)$$

□

We know that  $n$ -th order polynomials with real-valued coefficients always have  $n$  roots, but only if we allow those roots to be complex-valued. In general a polynomial with real coefficients cannot always be factored into a product of lower-order polynomials with real coefficients (for example  $X^2 + 1$ ). Similarly, a  $GF(2)$  polynomial cannot always be factored into two or more polynomials with  $GF(2)$  coefficients.

#### Example 12-34.

Continuing Example 12-33, the polynomial  $(1 \oplus D \oplus D^2)$  cannot be factored into the product of two first order polynomials over  $GF(2)$ . In fact, the only first-order polynomials over  $GF(2)$  are  $D$  and  $(1 \oplus D)$ , and the reader can readily verify that they cannot be factors of  $(1 \oplus D \oplus D^2)$ . □

A polynomial that has no factors other than itself and 1 is called an *irreducible polynomial over  $GF(2)$* . In the sequel we will assume that the generator polynomial  $h(D)$  is irreducible.

Returning to the feedback shift-register, the state  $(x_{k-1}, \dots, x_{k-n})$  can assume at most  $2^n$  distinct values. From this fact, and other properties of the register, we can

discern the following properties:

- If the state of the shift-register is all-zero ( $00 \cdots 0$ ) at any time, then it must always be all-zero. Thus, we must ensure that this state is never visited unless we are satisfied with a complicated circuit that just generates all-zeros at the output.
- If the state ever stays the same from one time increment to the next, then it will forever be the same. Thus, if the output is to be interesting (anything but all-zeros or all-ones), then we must ensure that the state always changes upon every time increment.
- The sequence of states must be periodic. Since there are only  $2^n$  distinct states, the sequence of states must always return to an initial state, after which the sequence of states repeats. Since the output  $x_k$  is a function of the state, it must also be periodic.
- Combining the first three, the maximum period of the states and outputs must be  $(2^n - 1)$  time increments. This maximum period would correspond to a periodic sequence of states which change at every time increment and which cycle through every state except the all-zero state.

A feedback shift-register is called *maximal-length* if the period of the output is  $r = 2^n - 1$ .

**Example 12-35.**

The generator polynomial for the shift-register of Example 12-30 is  $h(D) = 1 \oplus D \oplus D^2$ . From Example 12-34 this generator polynomial is irreducible. We can verify that the shift register is maximal-length, that is has period  $2^2 - 1 = 3$ . Starting with state (0,1), the following table specifies the state and output vs. time for four cycles:

$x_k$	$x_{k-1}$	$x_{k-2}$
1	0	1
1	1	0
0	1	1
1	0	1

Note that the state has returned to its initial value at the fourth time increment, and therefore the shift-register will continue with the same sequence of states. Also note that if we initialized the state with any of the other two values, the same sequence of states would result, but we would just start at a different point in the sequence. □

We could easily envision that the period of a shift-register sequence could be less than  $2^n - 1$  in length.

**Exercise 12-10.**

Show that the shift-register sequence corresponding to polynomial  $h(D) = 1 \oplus D^2$  has period one or two depending on the initial state. □

We would like to have some criterion to establish when a generator polynomial corresponds to a maximal-length shift-register sequence. When an irreducible

2	7	19	2000047
3	13	20	4000011
4	23	21	10000005
5	45	22	20000003
6	103	23	40000041
7	211	24	100000207
8	435	25	200000011
9	1021	26	400000107
10	2011	27	1000000047
11	4005	28	2000000011
12	10123	29	4000000005
13	20033	30	10040000007
14	42103	31	20000000011
15	100003	32	40020000007
16	210013	33	100000020001
17	400011	34	201000000007
18	1000201		

**Table 12-2.** Minimal weight primitive polynomials of orders two through 34[36]. Each entry in the table is an octal number, which when converted to binary specifies the coefficients of the polynomial  $h(D)$ . The most significant (left-most) bit is  $h_n = 1$  and the least significant (right-most) bit is  $h_0 = 1$ .

polynomial  $h(D)$  of degree  $n$  does not divide any polynomial  $(1 \oplus D^m)$  for  $m < 2^n - 1$ , it is said to be *primitive*. A shift-register sequence is maximal-length if and only if the generator polynomial is primitive [36].

**Example 12-36.**

We can verify that the generator polynomial of Example 12-30,  $h(D) = 1 \oplus D \oplus D^2$ , is primitive. This is because it obviously does not divide  $(1 \oplus D^2)$ , while it does divide  $(1 \oplus D^3)$  since

$$(1 \oplus D \oplus D^2)(1 \oplus D) = 1 \oplus D^3 \tag{12.80}$$

from Example 12-33.  $\square$

Fortunately, there exist primitive polynomials of all orders. The polynomials with *minimum weight*, that is with the minimum number of shift-register taps, of all orders up to  $n = 34$  are listed in Table 12-2.

**Example 12-37.**

A maximal-length shift-register of order 12 can be found from Table 12-2. The octal entry is "10123", which corresponds to binary "1000001010011" and hence polynomial

$$h(D) = 1 \oplus D \oplus D^4 \oplus D^6 \oplus D^{12} . \tag{12.81}$$

Hence the shift-register is characterized by difference equation

$$x_k = x_{k-1} \oplus x_{k-4} \oplus x_{k-6} \oplus x_{k-12} . \tag{12.82}$$

□

An interesting property of maximal-length sequences is that if we look at  $n$ -bit segments of the sequence, we will see all possible  $n$ -bit words, with the exception of the all-zero word. This follows from the fact that the state of the shift-register passes through all possibilities except all-zeros, and the state is equal to the past  $n$  bits of the output. The maximal-length sequence therefore satisfies a minimal condition for "randomness", since we would expect to see all combinations of bits (except the all-zero) in such a sequence.

The output of a maximal-length shift register is often called a *pseudorandom sequence*. This is because, even though the sequence is deterministic and periodic, it displays many of the properties of a random sequence (analogous for example to a numerical algorithm for random number generation). We can see these properties reflected in the *relative frequency* and in the *autocorrelation function*.

The relative frequency of observing particular sequences of  $i$  bits in a maximal-length sequence is close to the probability of observing the  $i$  bits in an i.i.d. random sequence as long as  $i \leq n$ , since all possible sequences of  $n$  bits occur once in one period of  $2^n - 1$  bits, with the exception of the all-zero sequence.

#### Exercise 12-11.

Show that the relative frequency of any particular sequence of  $i \leq n$  bits in the maximal-length sequence is

$$\frac{2^{n-i}}{2^n - 1} \approx 2^{-i} \quad (12.83)$$

for the case where the  $i$  bits do not constitute the all-zero sequence, and for the all-zero sequence of  $i$  bits

$$\frac{2^{n-i} - 1}{2^n - 1} \approx 2^{-i} \quad (12.84)$$

The approximations apply to large  $n$ , and hence for this case the sequence looks random on a relative frequency basis as long as we don't observe blocks of bits greater than  $n$ . □

The autocorrelation function can be determined using the *cycle-and-add property* of the maximal-length sequence [37]. This property says that if we modulo-two add the maximal-length sequence to itself, where one of the sequences has been shifted in time, we get another version of the same sequence shifted in time,

$$x_k \oplus x_{k+l} = x_{k+j} \quad (12.85)$$

for  $1 \leq l \leq r-1$ , where  $j$  depends on  $l$ . Of course, when  $l = 0$  the sum is the all-zero sequence (this is a degenerate case of a maximal-length sequence). The cycle-and-add property follows from the fact that if  $h(D)X(D) = 0$ , then obviously  $(1 \oplus D^l)h(D)X(D) = 0$ , and therefore  $(1 \oplus D^l)X(D)$  must also have generator polynomial  $h(D)$ . Many interesting properties can be derived from (12.85).

**Example 12-38.**

Since  $x_k \oplus x_{k+l} = 0$  if and only if  $x_k = x_{k+l}$ , it follows that  $x_k = x_{k+l}$  for precisely  $(r-1)/2$  values of  $k$  within one period  $0 \leq k \leq r-1$ , and  $x_k \neq x_{k+l}$  for precisely  $(r+1)/2$  values of  $k$ . Again,  $r = 2^n - 1$  is the length of the sequence.  $\square$

In terms of the autocorrelation, we are usually interested in the autocorrelation of a binary antipodal sequence  $s_k$  obtained by mapping  $x_k = 0$  into  $s_k = -1$  and  $x_k = 1$  into  $s_k = +1$ . We will call this new sequence the *binary antipodal maximal-length sequence*. The autocorrelation function of this sequence is defined as

$$R_s(l) = \frac{1}{r} \sum_{k=0}^{r-1} s_k s_{k+l} . \quad (12.86)$$

This is a *time-average* autocorrelation function averaged over one period of the sequence. Of course it is a periodic function of  $l$ , and hence we need only be concerned with the value for  $0 \leq l \leq r-1$ . Similarly, we can define a time-average mean value of the sequence as

$$\mu_s = \frac{1}{r} \sum_{k=0}^{r-1} s_k . \quad (12.87)$$

**Exercise 12-12.**

Using the relative frequency property, show that

$$\mu_s = \frac{1}{r} \quad (12.88)$$

which approaches zero as  $n$  (and hence  $r$ ) gets large.  $\square$

**Exercise 12-13.**

Use the cycle-and-add property to show that the autocorrelation function is given by

$$R_s(l) = \begin{cases} 1, & l = 0 \\ -\frac{1}{r}, & 1 \leq l \leq r-1 \end{cases} . \quad (12.89)$$

Hence, when  $r$  is large, the time-average autocorrelation function approaches zero except at multiples of the period. Except for the periodicity, this approaches the autocorrelation of a white sequence, and hence is another indication of the pseudo-random property.  $\square$

Using this time-average autocorrelation, we can infer another important property of the binary antipodal maximal-length sequence; namely, its harmonic structure. Since this sequence is periodic, we can expand it using a DFT,

$$s_k = \frac{1}{r} \sum_{m=0}^{r-1} S_m e^{j2\pi mk/r} , \quad (12.90)$$

where

$$S_m = \sum_{k=0}^{r-1} s_k e^{-j2\pi mk/r}, \quad 0 \leq m \leq r-1. \quad (12.91)$$

We can easily relate the harmonics of the sequence to the autocorrelation function.

### Exercise 12-14.

- (a) Show that

$$\sum_{k=0}^{r-1} s_{k+l} e^{-j2\pi mk/r} = e^{j2\pi ml/r} \sum_{k=0}^{r-1} s_k e^{-j2\pi mk/r}. \quad (12.92)$$

- (b) Show that

$$\frac{1}{r} |S_m|^2 = \sum_{l=0}^{r-1} R_s(l) e^{-j2\pi ml/r}. \quad (12.93)$$

- (c) Evaluate this DFT to show that

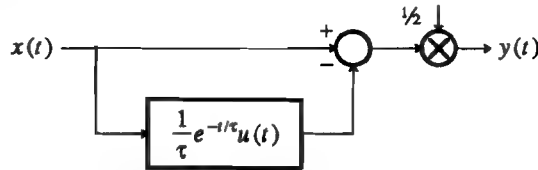
$$|S_m|^2 = \begin{cases} 1, & m = 0 \\ r+1, & 1 \leq m \leq r-1 \end{cases}. \quad (12.94)$$

Hence, the harmonics of the sequence are all equal to one another in magnitude, except for the d.c. component, which is relatively small. This resembles the power spectrum of a white sequence, and this property makes these sequences desirable as spreading sequences in spread spectrum (Section 8.6).  $\square$

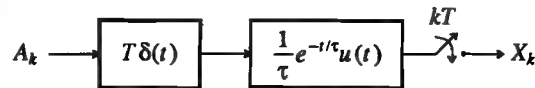
## PROBLEMS

- 12-1. Consider the a.c. coupled circuit in Figure 12-1a.

- (a) Show that an equivalent representation of the circuit in terms of a linear time-invariant system is shown below:



- (b) Assume an input PAM signal with delta-function pulses with area  $T$  (since real PAM pulses have width of order  $T$ ), yielding the equivalent system below (assuming symbol-rate sampling in the receiver):

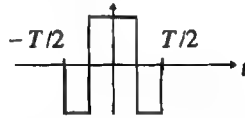


Calculate the output ISI, thereby establishing the equivalency of Figure 12-1b.

### 12-2.

- (a) What is the largest possible ISI in Figure 12-1b due to baseline wander, where no provision has been made to counter baseline wander in the line code?

- (b) Evaluate this maximum ISI for a.c. coupling cutoff frequency equal to 1% of the symbol rate.
- 12-3. Repeat Exercise 12-1 for the *Walz pulse shape*, a binary antipodal line code with the pulse shape shown below:



Evaluate the intersymbol interference for the same  $\beta$  as in Example 12-2.

- 12-4. Assume that a baseband PAM system uses a transmitted biphasic pulse, and through a combination of transmit filtering and receive equalization we equalize to a pulse shape that satisfies the Nyquist criterion.
- Show that such an equalized pulse must have an excess bandwidth greater than 100%. Thus, the minimum bandwidth for an equalized biphasic pulse shape is twice that required for transmitted pulses that are allowed to have a d.c. content.
  - Explicitly write the Nyquist criterion for an equalized biphasic pulse with excess bandwidth greater than 100% and less than 150%.
  - What is the equalized biphasic pulse shape with the minimum bandwidth that satisfies the Nyquist criterion?
  - Does the result of a. apply to all transmitted pulses which have zero area?
- 12-5. Keeping the average transmitted power the same, compare the noise immunity for a twinned binary code and a binary antipodal code. Assume the input information bits are independent, and that a "one" has probability  $p$ . Assume the same pulse shape in both cases, assume all time-translates of the basic pulse are orthogonal, and be sure to take into account the transmitted signal power and the fact that in the twinned binary code the transmitted levels are not equally likely.
- Show that for  $p = 1/2$ , the noise immunity of the binary antipodal code is 3 dB better.
  - Compare the noise immunity for the same average transmitted power for arbitrary  $p$ . For what range of  $p$  does the antipodal code have better noise immunity?
- 12-6. Interpret the twinned binary code as a binary antipodal transmitted data symbol with a transmitted pulse shape that is two symbol intervals wide.
- 12-7. For a twinned binary code, show how a DFE could be used in the receiver in place of the receiver structure of Figure 12-3. Compare the noise immunity of this approach to that of the conventional interpretation (Problem 12-5).
- 12-8. Show how the Viterbi algorithm can be used to decode the twinned binary code, and determine the advantage in  $\Xi_{VA}$ , the argument of  $Q(\cdot)$ , that can be obtained.
- 12-9. Show that the Viterbi algorithm can be used to advantage to decode an AMI-coded signal. You will have to make an assumption about the noise statistics at the AMI slicer input. What is the improvement in  $\Xi_{VA}$ , the argument of  $Q(\cdot)$ ?
- 12-10. In an AMI decoder, a *bipolar violation* is the term for a violation of the known constraints on sequences of ternary levels. The term arises because AMI line coding is sometimes called bipolar line coding.
- List all possible bipolar violations.
  - Describe how bipolar violations can be used for in-service monitoring.
  - Make a table that relates the number of bit errors to the number of bipolar violations for all single errors in slicing of the ternary signal. From this table estimate the relationship between the rate of bit errors and bipolar violations.

- 12-11. Adopt the following notation for a pseudoternary line code. A transmitted "0" is just that, a transmitted "B" is a non-zero symbol which obeys the AMI constraint (it is opposite in polarity to the last transmitted non-zero symbol), and a transmitted "V" is a non-zero symbol that violates this constraint (has the same polarity as the last transmitted non-zero symbol).
- Describe an AMI line code in these terms.
  - Give an example of a signal containing "V"'s that is still d.c. balanced. What price do we pay for introducing "V"'s?
- 12-12. Using the notation of Problem 12-11, the B6ZS (*bipolar six-zero substitution*) code substitutes at the output of an AMI coder, for each block of six "zeros", the code word "B0VB0V". Specifically then, the transmitted block of six symbols would be "+0+-0-" for an RDS of zero at the start of the block (last transmitted non-zero symbol was "-"), and "-0+0+" otherwise.
- What is the advantage of this?
  - Describe the decoder.
  - What is the RDS of this code?
- 12-13. An alternative to the B6ZS code of Problem 12-12 is the HDBk (*High-Density Bipolar*) code, which achieves a lower DSV. The code word "B00...0V" or "00...0V" is substituted for a block of  $k+1$  "zeros", where each of these code words is  $k+1$  symbols long. The appropriate code word is chosen so as to make the number of "B"'s between consecutive "V"'s odd. Note that the two code words allow us to put in a "B" or not, and hence we can control whether the number of "B"'s is even or odd by the choice of the code word. For example, in HDB3, if the number of "B"'s since the last "V" is even at the beginning of the block of four "zeros", then we transmit block "B00V". Otherwise, we transmit "000V". Show that the RDS of HDBk is limited to the range  $-1 \leq \text{RDS} \leq +1$ , and hence the DSV is two. The penalty relative to AMI in baseline wander is therefore small. (HINT: Consider the disparity of a block of symbols starting at one "V" and extending up to the next "V".)
- 12-14.
- Make a reasonable definition of the code B4ZS (see Problem 12-12).
  - What is the RDS and DSV of this code?
  - What advantages or disadvantages might this code have over B6ZS?
- 12-15.
- Show that there is no B3ZS code similar to B6ZS defined in Problem 12-12.
  - Show that by introducing two modes into the code, a B3ZS code can be defined.
- 12-16. For the 4B3T line code described in this chapter, describe how in-service monitoring of error rate could be performed at the decoder.
- 12-17. Define the *one's density* of a pseudoternary code as follows: for each  $n \geq 1$  it is the smallest value the quantity  $\frac{m}{n}$  can assume, where  $n$  is the number of symbols in a block and  $m$  is the number of non-zero symbols in this block. Plot this quantity for AMI, B6ZS, HDB3, and 4B3T.
- 12-18. Design a bimode binary block code which maps three information bits into four transmitted binary data symbols (75% efficiency) and maintains a DSV less than or equal to four.
- 12-19. Modify the results of Example 12-15 to yield a first-order spectral null at  $z = -1$ , half the sampling rate.
- 12-20. Show by example that if the ternary slicer in Figure 12-12c makes an error, this error can propagate. Under what conditions is this propagation the biggest problem?
- 12-21. *Class II partial response.* Given a partial response system with polynomial  $F(D) = (1 + D)^2$ .
- Draw a typical transmitted pulse shape and describe qualitatively what is accomplished by using this shape.



- (b) Give a truth table for the precoder and give a Boolean logic expression for the precoder design.
  - (c) How many levels does the slicer have? Specify the decoder.
- 12-22. Let the input bits be independent and identically distributed, with the probability of a "one" equal to  $p$ . The power spectrum for the duobinary PR output symbols was determined (equivalently for AMI) in Section 12.2. Determine this output spectrum for duobinary and modified duobinary.
- 12-23. Given a discrete-time channel given by

$$Y_k = X_k + \rho X_{k-1} + N_k \quad (12.95)$$

where  $0 < \rho < 1$  and the additive noise is white and zero-mean with variance  $\sigma^2$ , derive the error probability or bounds on the error probability expressed in terms of both the peak and average transmitted power at the transmitter output for the following cases:

- (a) We use *receiver LE-ZF equalization* and binary antipodal signaling.
  - (b) Same as a., except we use *transmitter equalization*.
  - (c) We use *duobinary PR precoding* in the transmitter, and in the receiver we equalize to a  $(1 + D)$  response prior to the three-level slicer.
  - (d) Same as c., except we do the equalization in the transmitter.
  - (e) We use binary antipodal signaling together with ML sequence estimation in the receiver.
- 12-24. Specify a general scheme to use the redundancy inherent in a PR encoded signal to do performance monitoring (unreliable error detection) at the slicer output. Specify this scheme specifically for duobinary and duobinary PR with a three-level slicer, and relate to earlier results in Section 12.1.
- 12-25. Generalize PR from the binary case considered in the chapter to an input PAM signal with  $M$  equally-spaced levels.
- 12-26. Consider the MSK signal in (12.57).
- (a) Using Figure 12-17 as a starting point, draw a trellis with a finite number of states that describes the phase evolution of the MSK signal.
  - (b) Show that the minimum-distance error event is the error event of length one. Find its distance.
  - (c) Compare the optimal sequence detector performance to that of the receiver in Figure 6-46.
- 12-27. For a frame-synchronized scrambler, find a pathological input bit stream with period equal to  $r$  that results in a scrambled sequence with period two.
- 12-28. Use Table 12-2 to design a maximal-length shift-register of order  $n = 3$ . Calculate the sequence of states and outputs to verify that the period is  $2^3 - 1 = 7$ .
- 12-29. Repeat Problem 12-28 for  $n = 4$ .

## REFERENCES

1. P. A. Franaszek, "Sequence-State Coding for Digital Transmission," *BSTJ* 47 p. 143 (Jan. 1968).
2. A. R. Calderbank and J. E. Mazo, "Spectral Nulls and Coding with Large Alphabets," *IEEE Communications Magazine*, (Dec. 1991).
3. S. Yoshida and S. Yajima, "On the Relationship Between Encoding Automaton and the Power Spectrum of its Output Sequence," *Trans. IECE (Japan)* E59 p. 97 (1976).
4. N. Q. Duc and B. M. Smith, "Line Coding for Digital Data Transmission," *Australian Telecommunications Research (A. T. R)* 11(2)(1977).
5. A. Croisier, "Introduction to Pseudoternary Transmission Codes," *IBM J. Research and Development* 14 p. 354 (July 1970).
6. A. Brosio, U. DeJulio, V. Lazzari, R. Ravaglia, and A. Tofanelli, "A Comparison of Digital Subscriber Line Transmission Systems Employing Different Line Codes," *IEEE Trans. on Communications* COM-29(11) p. 1581 (Nov. 1981).
7. L. A. Meacham, "Twinned Binary Transmission," *U.S. Patent* 2,759,047, 0.
8. M. R. Aaron, "PCM Transmission in the Exchange Plant," *BSTJ* 41 pp. 99-141 (Jan. 1962).
9. H. Sailer, H. Schenk, and E. Schmid, "A VLSI Transceiver for the ISDN Customer Access," *Proc. IEEE Int. Conf. Communications*, (June 1985).
10. R. F. Lyon, "Two-Level Block Encoding for Digital Transmission," *IEEE Trans. on Communications* COM-21(12) p. 1438 (Dec. 1973).
11. J. N. Franklin and J. R. Pierce, "Spectra and Efficiency of Binary Codes without DC," *IEEE Trans. on Communications* COM-20(6) p. 1182 (Dec. 1972).
12. J. K. Wolf, "Modulation and Coding for the Magnetic Recording Channel," *Proceedings NATO Advanced Study Institute*, (July 1986).
13. H. Kobayashi, "A Survey of Coding Schemes for Transmission or Recording of Digital Data," *IEEE Trans. on Communications* COM-19 p. 1087 (Dec. 1971).
14. G. D. Forney, Jr and A. R. Calderbank, "Coset Codes for Partial Response Channels; Or Coset Codes with Spectral Nulls," *IEEE Trans. on Information Theory* IT-35 p. 925 (1989).
15. A. R. Calderbank and J. E. Mazo, "Baseband Line Codes Via Spectral Factorization," *IEEE Trans. on Selected Areas of Communications*, p. 914 (Aug. 1989).
16. D. G. Messerschmitt, "Generalized Partial Response for Equalized Channels with Rational Spectra," *IEEE Trans. on Communications* COM-23(11) p. 1251 (Nov. 1975).
17. A. Lender, "The Duobinary Technique for High-Speed Data Transmission," *IEEE Trans. on Commun. Electronics* 7(March 1963).
18. E. Kretzmer, "Generalization of a Technique for Binary Data Communication," *IEEE Trans. on Communication Tech.* COM-14(Feb. 1966).
19. D. G. Messerschmitt, "Design of a Finite Impulse Response for the Viterbi Algorithm and Decision Feedback Equalizer," *Proc. IEEE Int. Conf. on Communications*, (June 1974).
20. T. Aulin, N. Rydbeck, and C.-E. W. Sundberg, "Continuous Phase Modulation — Part II: Partial Response Signaling," *IEEE Trans. on Communications* COM-29(3)(March 1981).
21. F. D. Waldhauer, "Quantized Feedback in an Experimental 280-Mb/s Digital Repeater for Coaxial Transmission," *IEEE Trans. on Communications* COM-22(1) p. 1 (Jan. 1974).
22. J. E. Savage, "Some Simple Self-Synchronizing Digital Data Scramblers," *BSTJ* 46(2) p. 449 (Feb. 1967).

23. D. G. Leeper, "A Universal Digital Data Scrambler," *BSTJ* 52(10) p. 1851 (Dec. 1973).
24. S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1987).
25. P. H. Siegel and J. K. Wolf, "Modulation and Coding for Information Storage," *IEEE Communications Magazine*, (Dec. 1991).
26. P. Kabal and S. Pasupathy, "Partial-Response Signaling," *IEEE Trans. on Communications* COM-23(9)(Sep. 1975).
27. S. Pasupathy, "Correlative Coding: Baseband and Modulation Applications," pp. 429 in *Advanced Digital Communications Systems and Signal Processing Techniques*, ed. K. Feher, Prentice-Hall, Englewood Cliffs, N.J. (1987).
28. H. Kobayashi, "Correlative Level Coding and Maximum-Likelihood Decoding," *IEEE Trans. Inform. Theory* IT-17 pp. 586-594 (Sept. 1971).
29. G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. on Information Theory* IT-18 pp. 363-378 (May 1972).
30. T. Aulin and C.-E. W. Sundberg, "Continuous Phase Modulation — Part I: Full Response Signaling," *IEEE Trans. on Communications* COM-29(3)(March 1981).
31. M. K. Simon, "A Generalization of Minimum-Shift-Keying (MSK)-Type Signaling Based Upon Input Data Symbol Pulse Shaping," *IEEE Trans. on Communications* COM-24(8)(Aug. 1976).
32. J. B. Anderson, C.-E. W. Sundberg, T. Aulin, and N. Rydbeck, "Power-Bandwidth Performance of Smoothed Phase Modulation Codes," *IEEE Trans. on Communications* COM-29(3)(March 1981).
33. S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Communications Magazine* 17(4)(July 1979).
34. S. Haykin, *Communication Systems*, 2nd Edition, John Wiley & Sons, Inc. (1983).
35. R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic Publishers, Norwell, Mass. (1987).
36. W. Peterson and E. Weldon, *Error-Correcting Codes*, 2nd Ed., M.I.T. Press, Cambridge, Mass (1972).
37. S. W. Golomb, *Shift Register Sequences*, Holden-Day, San Francisco (1967).

# 13

---

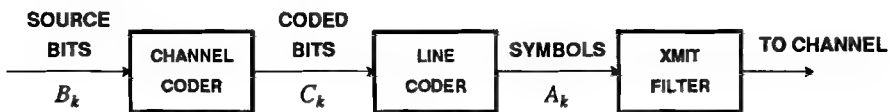
## ERROR CONTROL

---

Chapter 12 concentrated on coding techniques for controlling the power spectrum of the transmitted signal. A side benefit is often the capability for limited error detection.

A *channel coder* precedes the line coder as shown in Figure 13-1, and is designed specifically for one of three purposes:

- *Error detection* can be used to increase reliability. For example, a code can be devised that will detect *all* single bit errors, not just a fraction of them as in line coding. Error detection can ensure reliability by causing the retransmission of blocks of data until they are correctly received. In-service monitoring and error detection are similar, except that with in-service monitoring it is not necessary to detect all or even the majority of errors, since all we need is an indication of the



**Figure 13-1.** A channel coder translates source bits into coded bits to achieve error detection, correction, or prevention.

performance of the system rather than the location of each and every error.

- A more ambitious goal is *error correction*. This carries error detection one step further by actually using the redundancy to correct transmission errors.
- Instead of correcting errors after they occur, a still more ambitious goal is *error prevention*. The probability of error is reduced by combining detection and decoding to get a technique known as *soft decoding*.

There are two broad classes of techniques for introducing redundancy into a signal, both of which can be used simultaneously. Suppose that transmitting without redundancy requires the symbol rate  $f_{b,0}$  and alphabet size  $L_0$ , yielding the bit rate  $B = f_{b,0} \log_2 L_0$ . The first method, covered in this chapter, is to increase the symbol rate, making  $f_b > f_{b,0}$ , by inserting extra transmitted symbols which depend deterministically on the information bits.

#### Example 13-1.

In a *binary block code*,  $k$  information bits are combined with  $n-k$  extra redundant bits to yield a block of  $n$  bits which are transmitted as binary data symbols. This block code increases the symbol rate by a factor of  $n/k$ . The redundancy can be used for error detection, correction, and/or prevention.  $\square$

The second method, covered in Chapter 14, is to increase the number of symbols in the alphabet,  $L > L_0$ , resulting in a so-called *signal-space code*. Signal-space codes combine the channel and line coding operation into one.

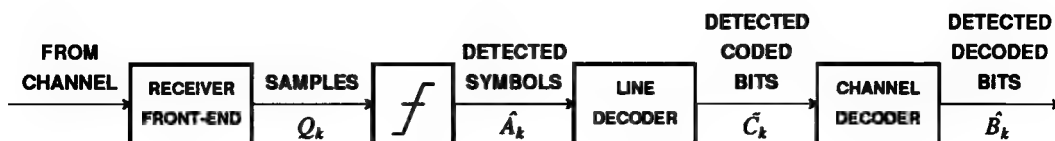
#### Example 13-2.

While keeping the symbol rate equal to the bit rate, we can transmit a signal using an alphabet with four symbols instead of two. This implies that each information bit must be mapped into one of four symbols. If this is done correctly, the receiver can use the redundant information for error control.  $\square$

Increasing the size of the alphabet is usually the most advantageous method on bandlimited media such as radio or voiceband data channels, as increasing the symbol rate also increases the bandwidth and often lets more noise into the detector. For this reason, the signal-space codes of Chapter 14 are most often used on such media. Increasing the symbol rate is acceptable on media that provide abundant bandwidth, such as optical fiber.

Two fundamentally different types of *decoding* are used, *hard* and *soft*. Hard decoding is the easiest to understand, and is illustrated in Figure 13-2. A slicer makes a "hard decision", doing its best to detect the transmitted symbols (complete with their redundancy). Redundancy is then removed by inverting the mapping performed in the encoder. Since not all bit patterns are permitted by the code, the decoder can *detect* or *correct* bit errors. From the perspective of the coder, the channel is binary and makes transmission errors. Often a binary symmetric channel (BSC) noise generation model is used (Figure 9-2).

A soft decoder, by contrast, makes direct decisions on the information bits without making intermediate decisions about the transmitted symbols. Soft decoding starts with the continuous-valued samples of the received signal, processing these

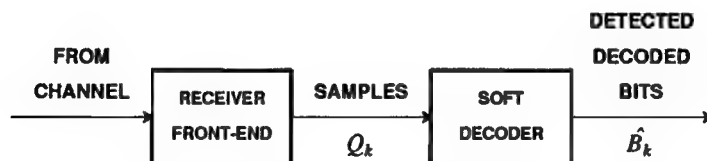


**Figure 13-2.** In hard decoding of a channel code, decisions are made about the incoming symbols by a slicer, the symbols are decoded into bits by a line decoder, and the channel decoder maps these coded bits into uncoded bits.

directly to detect the decoded bit sequence, as shown in Figure 13-3. Instead of correcting errors after they occur, as done by a hard decoder, a soft decoder *prevents* errors by combining slicing and line decoding with channel decoding. We can think of soft decoding as a combination of slicing and removing redundancy.

Soft decoding is capable of providing better performance, at the expense of implementation complexity, since it makes use of information that the slicer would otherwise throw away. For example, if the slicer input is halfway between slicer levels, the hard decoder would be forced to make a decision, whereas the soft decoder would make note of the uncertainty and incorporate that uncertainty in the final decision. Soft decoding is an integral part of the demodulation process, and therefore hard decoding is our only option when we have no control over the modulation and demodulation process.

We will consider two broad classes of channel codes, *block codes* and *convolutional codes*. Both hard and soft decoding are used for block and convolutional codes, while only soft decoding is used for signal-space codes (Chapter 14). Chapter 12 applied block codes to line coding. The principle is the same for channel coding; blocks of  $k$  source bits are mapped into blocks of  $n$  coded bits where  $n > k$ . Such a block code is said to have *code rate*  $k/n$ , where the terminology refers to the fraction of the total bit rate devoted to information bits. A convolutional coder also produces coded bits at a higher rate than the source bits, but it does so without dividing the source bits into blocks. In both cases there are more coded bits than source bits, and the coded bits have redundant information about the source bits.



**Figure 13-3.** A soft decoder operates directly on samples of the incoming signal rather than on detected bits, and often the additive Gaussian noise generation model is used (Example 9-14).

For each channel code we will compare the performance of an uncoded system with a coded system with both hard and soft decoders. This is done by considering the SNR required at the detector input to achieve a fixed probability of error. The coded system can tolerate a lower SNR than the uncoded system; this difference (in dB) is called the *coding gain*. The coding gain can be viewed as a decrease in the signal power allowable in the coded system for a fixed noise power, or an increase in the allowable noise power for a fixed signal power. The coding gain is not the full story, however.

### Example 13-3.

Consider a white Gaussian noise channel. By keeping the line code fixed and increasing the symbol rate to accommodate redundancy, we increase the bandwidth of the system and admit more noise into the detector. Useful codes must have enough coding gain to more than compensate for this extra noise.  $\square$

### Example 13-4.

When Gaussian noise is bandlimited to the same bandwidth as the signal, and has fixed power, as is usually assumed for a jammer (Section 8.5), increasing the bandwidth to accommodate redundancy has no effect on the noise at the detector. In this case, the full advantage of the coding gain can be realized.  $\square$

These examples illustrate that coding of the type discussed in this chapter is more beneficial in the same situations where spread spectrum is appropriate (Section 8.5). In fact, channel coding is usually considered an inherent part of spread spectrum modulation. Where increased bandwidth results in increased noise, the signal space codes of Chapter 14 are usually more appropriate.

Given a digital communication system with a bit stream at the input and output, without modifying the medium or modulation technique we can add coding as shown in Figure 13-4. In other words, a digital communication system has been designed and implemented, the error rate is too great for our intended purpose, and so we decrease the error rate at the expense of a lower information rate. Since hard decoding must be used, signal-space codes cannot be considered.

The emphasis in this book will be on soft decoding, because it is closely tied into the modulation and demodulation process, and because it gives better performance. Error correction coding is a large subject, and in this book we hope to convey the most important concepts and leave the details to the extensive and excellent literature on



Figure 13-4. A channel coder and hard decoder added to an existing digital communication system.

the subject.

## 13.1. BLOCK CODES

An  $(n, k)$  block coder maps blocks of  $k$  source bits into blocks of  $n$  coded bits. The  $n$  coded bits depend only on the  $k$  source bits, so the coder is said to be *memoryless*. Assume that the  $k$  source bits are collected in a shift register, as shown in Figure 13-5, and the  $n$  coded bits are serialized to be sent to the line coder. The block coder circuitry itself often consists only of modulo-two adders (exclusive-or gates).

### Example 13-5.

In a *single-parity-check code*, the number of ones in a coded block must be even. The first  $k$  bits of the codeword are the same as the source bits, or in the notation of Figure 13-5,

$$C^{(1)} = B^{(1)}, \dots, C^{(k)} = B^{(k)}. \quad (13.1)$$

An extra bit is appended to ensure that there are an even number of ones in  $(C^{(1)}, \dots, C^{(n)})$ , which occurs if and only if  $C^{(1)} \oplus \dots \oplus C^{(n)} = 0$ , where  $\oplus$  denotes modulo-two summation. The extra bit should therefore be

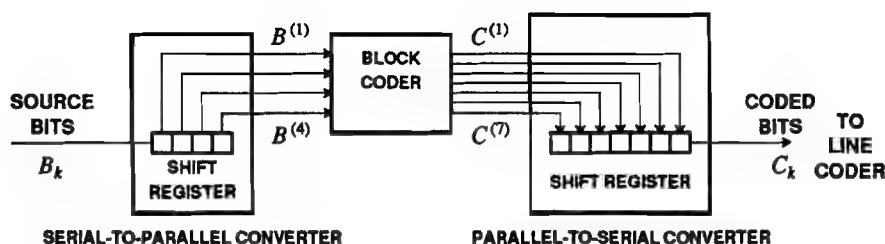
$$C^{(n)} = C^{(k+1)} = B^{(1)} \oplus \dots \oplus B^{(k)}. \quad (13.2)$$

□

In a *general parity-check code*, all codewords are modulo-two summations of subsets of the source bits. A parity-check code is fully characterized by a *generator matrix*  $G$ , which consists only of zeros and ones. If the input and output bits are collected into the *row* vectors  $\mathbf{b}$  and  $\mathbf{c}$ , then

$$\mathbf{c} = \mathbf{b}G \quad (13.3)$$

where the addition that occurs in the matrix multiplication is modulo-two.



**Figure 13-5.** A block coder collects  $k$  incoming bits in a shift register, codes them, and shifts them out serially to the line coder. The example shown is a  $(n, k) = (7, 4)$  block code.



**Example 13-6.**

The generator matrix for the single-parity-check code of Example 13-5 is

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \end{bmatrix}. \quad (13.4)$$

□

Codes like that of Example 13-6 which have a generator matrix of the form

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}], \quad (13.5)$$

where  $\mathbf{I}_k$  is the  $k$ -dimensional identity matrix, are called *systematic*. The first  $k$  output bits  $C_1 \cdots C_k$  are exactly the input bits  $B_1 \cdots B_k$ . The single-parity-check code of Example 13-6 is systematic.

**Example 13-7.**

A more elaborate systematic parity-check code is the (7,4) *Hamming code*, which has generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (13.6)$$

□

For a particular binary row vector  $\mathbf{b}$ ,  $\mathbf{c} = \mathbf{bG}$  is called a *codeword*. A *code* is the set of all possible codewords. Every codeword is a modulo-two summation of rows of the generator matrix (the zero vector is a degenerate example). Clearly, therefore, the modulo-two sum of any two codewords is a codeword, and parity-check codes are said to be *closed* under modulo-two summation. This is a desirable property for codes, and is elaborated in Appendix 13-A. In fact, it is shown in the appendix that all parity-check codes are *linear*, and that all linear block codes are parity-check codes with some generator matrix. Furthermore, all linear block codes are equivalent to a systematic code, obtained by reordering the coded bits. In the following we specialize to linear codes and exploit their special properties.

Consider the performance of block codes, as measured by their coding gain, for both soft and hard decoding. An important goal is to demonstrate that soft decoding is superior. The basic difference between the two cases is that in the soft decoding case the detector chooses the codeword closest to the reception in Euclidean distance, and in the hard decoding case the detector uses Hamming distance. Furthermore, we will find, not unexpectedly in view of the analysis in Section 9.6, that the probability of error is dominated by the two codewords that are closest in Euclidean or Hamming distance. To compare the two approaches, we need to find a relationship between

Euclidean and Hamming distance between a given pair of codewords; this depends on the line code, that is, the mapping from coded bits  $C_k$  into symbols  $A_k$  in Figure 13-1. The Euclidean distance is measured in terms of the symbols, whereas the Hamming distance is measured in terms of coded bits. Comparisons below assume that binary antipodal signaling is used, so that the line code maps  $\{0,1\}$  into  $\{\pm a\}$ .

### Exercise 13-1.

Show that for a binary antipodal line code, the Hamming distance and Euclidean distance between a pair of codewords are related by

$$d_E = 2a \sqrt{d_H} \quad (13.7)$$

□

When the code is linear, it is shown in Appendix 13-A that the minimum Hamming distance between codewords is equal to the minimum Hamming weight (number of ones) among all the nonzero codewords,

$$d_{H,min} = \min_{\substack{c \in C \\ c \neq 0}} w_H(c) . \quad (13.8)$$

### Example 13-8.

In the single-parity-check code of Example 13-5, all codewords have an even-valued Hamming weight. Thus, the smallest Hamming weight among all nonzero codewords is two. This is also the minimum Hamming distance between codewords. Thus, the minimum Euclidean distance for binary antipodal signaling is  $d_{E,min} = 2a\sqrt{2}$ . □

## 13.1.1. Performance of Soft Decoders

For the soft decoding case, the input to the decoder is the sample stream before it is applied to a slicer. Assume the equivalent channel is a discrete-time additive Gaussian noise channel with independent noise components and no ISI. A codeword  $c$  is transmitted as a vector  $\mathbf{a}$  with binary antipodal components, and the noise samples have variance  $\sigma_c^2$  where the subscript refers to the coded system. The received samples  $Q_k$  can be collected into the vector

$$\mathbf{q} = \mathbf{a} + \mathbf{n} \quad (13.9)$$

where  $\mathbf{n}$  is a vector of i.i.d. Gaussian random variables. This detection problem is familiar from Section 9.2, where we showed that the ML detector selects the vector  $\hat{\mathbf{a}}$  in  $\Omega_a$  closest in Euclidean distance to the observed vector  $\mathbf{q}$ . Hence,

$$\Pr[\text{block error}] \geq Q \left[ \frac{d_{E,min}}{2\sigma_c} \right] \quad (13.10)$$

where  $d_{E,min}$  is the minimum Euclidean distance between the transmitted vector  $\mathbf{a}$  and all other vectors in  $\Omega_a$ . If there are few codewords at distance  $d_{E,min}$  from the transmitted codeword, then the probability of error will be close to this bound. A crude upper bound on the number of codewords of distance  $d_{E,min}$  from any codeword

is  $2^k - 1$ , the number of other codewords in the code. Hence, an upper bound on the block error probability is

$$\Pr[\text{block error}] \leq (2^k - 1)Q\left[\frac{d_{E,min}}{2\sigma_c}\right], \quad (13.11)$$

where we have assumed that all the other codewords are at the closest distance  $d_{E,min}$  and have used the union bound of Sections 8.3.

**Example 13-9.**

In the single-parity-check code of Example 13-5, in view of Example 13-8,

$$Q\left[\frac{a\sqrt{2}}{\sigma_c}\right] \leq \Pr[\text{block error, soft decoding}] \leq (2^k - 1)Q\left[\frac{a\sqrt{2}}{\sigma_c}\right]. \quad (13.12)$$

□

To determine the coding gain, we must compare with an uncoded binary antipodal system. We can either determine the probability of block error for the uncoded system (even though it doesn't use blocks) or the probability of information bit error for the coded system. The probability of bit error is more frequently of interest. The uncoded system will have bit error probability

$$\Pr[\text{bit error, uncoded system}] = Q\left[\frac{a}{\sigma_u}\right] \quad (13.13)$$

for an alphabet of  $\pm a$ , where  $\sigma_u^2$  is the Gaussian noise variance at the slicer input for the uncoded system (note that depending on the medium, modulation technique, and bit rate,  $\sigma_u$  and  $\sigma_c$  may not be equal). We can bound the probability of bit error for the coded system. Decoding each block produces  $k$  bits, some of which may be in error if the block is in error. Each block error will produce at least one bit error, so we can write

$$\Pr[\text{bit error, soft decoding}] \geq \frac{1}{k} \Pr[\text{block error, soft decoding}]. \quad (13.14)$$

Another way to understand this claim is to observe that if one out of  $M$  blocks is in error, then at least one out of  $kM$  decoded bits will be in error. Furthermore, each block error cannot produce more than  $k$  bit errors, so

$$\Pr[\text{bit error, soft decoding}] \leq \Pr[\text{block error, soft decoding}]. \quad (13.15)$$

In words, if one out of  $M$  blocks is in error, then at most one of  $M$  bits will be in error.

**Example 13-10.**

For the  $(n, n-1)$  single-parity-check code (Example 13-5), we can make the probabilities in (13.10) and (13.11) approximately equal by setting the arguments of  $Q(\cdot)$  equal, that is, ignoring constant multipliers. This strategy yields

$$a_c\sqrt{2}/\sigma_c = a_u/\sigma_u \quad (13.16)$$

where  $a_c$  and  $a_u$  are the antipodal signal levels in the two cases. Ignoring the constant multiplier is suspect for large  $n$ , but let us forge ahead anyway. Defining the SNR to be the signal squared divided by the noise variance, we see that  $SNR_u = 2 SNR_c$ , or the coded system can have a lower SNR by  $10 \log_{10} 2 = 3$  dB. This is an estimate of the coding gain. If the channel noise  $\sigma$  is independent of the bandwidth of the signal, as for a fixed power interference, then this coding gain represents accurately the difference between the coded and uncoded systems.  $\square$

To fairly compare the coded and uncoded systems, we must use the same continuous-time channel in both cases, and take into account the possibly different noise variance that appears at the slicer for the two cases. The comparison is then between the signal powers required (not the SNR) for this fixed channel at the same error probability.

### Example 13-11.

Assume the channel is an additive white Gaussian noise channel. The coding gain of Example 13-10 doesn't tell the whole story, since it ignores the larger bandwidth, and hence greater noise, of the coded system. For the  $(n, n-1)$  single-parity-check code the coded system has symbol rate  $n/(n-1)$  larger than the uncoded system, so the bandwidth will be larger by the same factor, and

$$\sigma_c^2 = \frac{n}{n-1} \sigma_u^2. \quad (13.17)$$

Combining (13.17) with (13.16),

$$\frac{a_c^2}{a_u^2} = 0.5 \frac{n}{n-1}. \quad (13.18)$$

The signal power required for the coded system for  $n=3$  is 1.25 dB smaller than the uncoded system over the same white noise channel. The coding gain is 3 dB, but 1.75 dB of this is lost due to the larger noise on the coded channel.

We have ignored constant multipliers of  $Q(\cdot)$ . A more refined comparison, which retains the constant multipliers is as follows. Our lower bound on bit error probability for the coded system is

$$\Pr[\text{bit error, soft decoding}] \geq \frac{1}{n-1} Q \left[ \frac{a \sqrt{2(n-1)/n}}{\sigma_u} \right]. \quad (13.19)$$

Similarly, combining (13.17) with (13.15) and (13.11) we get

$$\Pr[\text{bit error, soft decoding}] \leq (2^{n-1} - 1) Q \left[ \frac{a \sqrt{2(n-1)/n}}{\sigma_u} \right]. \quad (13.20)$$

More precise comparisons require graphical or numerical techniques. In particular, the comparison of required signal power becomes a function of the SNR (it is not constant). For a given probability of error, we can numerically find the power advantage assuming the best case (13.19) and the worst case (13.20) (i.e. first assuming (13.19) is a tight bound, then assuming (13.20) is a tight bound). For the (3,2) single-parity-check code we tabulate the power advantage at a probability of error of  $10^{-5}$  and  $10^{-7}$  (see Problem 13-1):

probability of error	best case	worst case	ignoring constant
$10^{-5}$	1.56 dB	0.77 dB	1.25 dB
$10^{-7}$	1.45 dB	0.91 dB	1.25 dB

Notice that the bounds get closer together as the SNR increases. Eventually they will converge to the figure arrived at by ignoring the constant in front of the  $Q(\cdot)$ .  $\square$

This example illustrates a general technique for comparison. First find the coding gain, which relates the required SNR at the detector input. Then find the relative noise variances at the detector input for the coded and uncoded systems transmitting over the same channel; from these compare the required signal levels at the detector input.

### Example 13-12.

In the (7,4) Hamming code of Example 13-7, the minimum Hamming weight of all nonzero codewords is three (as is easily seen by considering linear combinations of rows of the generator matrix). Hence, from (13.7),

$$d_{E,min} = 2a\sqrt{3}. \quad (13.21)$$

Combining (13.14), (13.10), (13.15), and (13.11) we get

$$\frac{1}{4}Q\left[\frac{a\sqrt{3}}{\sigma_c}\right] \leq \text{Pr}[\text{bit error, soft decoding}] \leq 15Q\left[\frac{a\sqrt{3}}{\sigma_c}\right]. \quad (13.22)$$

Ignoring constant multipliers, the coding gain is therefore  $10\log_{10}3 = 4.7$  dB. The (7,4) Hamming code has symbol rate 7/4 times that of the uncoded system so on an additive white noise channel

$$\sigma_c = \sqrt{\frac{7}{4}}\sigma_u. \quad (13.23)$$

and (13.22) becomes

$$\frac{1}{4}Q\left[\frac{a\sqrt{12/7}}{\sigma_u}\right] \leq \text{Pr}[\text{bit error, soft decoding}] \leq 15Q\left[\frac{a\sqrt{12/7}}{\sigma_u}\right]. \quad (13.24)$$

Again, ignoring the constants in front of  $Q(\cdot)$  we get the approximate advantage in signal level for the coded system of,

$$10\log\frac{12}{7} = 2.34 \text{ dB} \quad (13.25)$$

which is much smaller than the coding gain. We can again numerically approximate the best and worst case power advantage corresponding to the bounds in (13.24),

probability of error	best case	worst case	ignoring constant
$10^{-5}$	3.00 dB	1.25 dB	2.34 dB
$10^{-7}$	2.79 dB	1.56 dB	2.34 dB

More precise comparisons can be made. However, it is clear that the comparison obtained ignoring the constant in front of the  $Q(\cdot)$  is accurate to within a fraction of a dB for

reasonable probabilities of error.  $\square$

The soft decoder can be expensive to implement because computing the distance between the observed vector and each possible codeword is usually impractical for large  $n$ . Fortunately, practical algorithms have been developed [1,2,3,4,5]. Also to be found in the literature are numerous bounds tighter than (13.11) (see for example [6]).

### 13.1.2. Performance of Hard Decoders

For a hard decoder, the decoding is done after the slicer. If the discrete-time channel up to the slicer has a noise generation model with independent noise components, then after the slicer the equivalent binary channel will usually be a BSC (Figure 9-2) with some error probability  $p$ . Let  $\mathbf{c}$  denote the transmitted codeword (a random vector of  $n$  bits) and let  $\tilde{\mathbf{c}}$  denote the corresponding bits emerging from the BSC. In this case we showed in Section 9.2 that the ML detector selects the codeword  $\hat{\mathbf{c}}$  closest in Hamming distance to  $\tilde{\mathbf{c}}$ .

#### Example 13-13.

For the (7,4) Hamming code, if  $\mathbf{c} = 0000000$  is transmitted and  $\tilde{\mathbf{c}} = 0001010$  is received, the ML detected codeword is  $\hat{\mathbf{c}} = 0001011$ , which is closer in Hamming distance than the all-zero codeword.  $\square$

The question that arises now is how many bit errors can be corrected by an ML detector for a given code. It is clear that if  $\tilde{\mathbf{c}}$  is closer to  $\mathbf{c}$  than to any other codeword then any errors in  $\tilde{\mathbf{c}}$  will be corrected by the ML detector. Certainly if  $\tilde{\mathbf{c}}$  has fewer than

$$t = \lfloor (d_{H,min} - 1)/2 \rfloor \quad (13.26)$$

errors then those errors can be corrected, where  $\lfloor \cdot \rfloor$  denotes the "floor" function, or the greatest integer less than or equal to the argument. This value  $t$  appeared before (see (9.24)).

#### Example 13-14.

In the single-parity-check code of Example 13-5 and Example 13-6, each codeword has an even number of ones, so  $d_{H,min} = 2$ . The code can correct

$$t = \lfloor (d_{H,min} - 1)/2 \rfloor = 0 \quad (13.27)$$

bit errors. Hence this code is not useful at all for hard error correction. It can detect any odd number of bit errors. Note that with soft decoding, this code is useful for reducing the error rate at a given signal level, but not with hard decoding.  $\square$

#### Example 13-15.

The (7,4) Hamming code has  $d_{H,min} = 3$  and  $t = 1$ , and hence can correct all single bit errors.  $\square$

In Section 9.2 we derived upper and lower bounds on the probability of error for vectors of bits transmitted over a BSC. Recall from Section 9.2 that these bounds are approximately equal when  $p$  is small and every codeword has exactly one other

codeword at distance  $d_{\min}$ . However, for practical block codes, this is rarely the case, so these bounds give only a rough estimate of the probability of block error. For some codes we can get better estimates.

For any code we can be *certain* of correcting up to  $t$  bit errors, but *some* error patterns with more bit errors may be correctable also, unless the code is a so-called *perfect* code. A perfect code has two properties:

- All bit patterns of length  $n$  are within Hamming distance  $t$  of a codeword, and
- No bit pattern of length  $n$  is Hamming distance  $t$  or less from more than one codeword.

**Example 13-16.**

The (7,4) Hamming code is a perfect code, so all seven-bit patterns are either a codeword or one bit distant from exactly one codeword. This implies that if  $c$  is transmitted and two bit errors occur,  $\tilde{c}$  will be distance one from some codeword  $\hat{c} \neq c$ , and a decoding error will occur.  $\square$

Perfect codes are optimal on the BSC in the sense that they minimize the probability of error among all codes with the same  $n$  and  $k$ . But very few exist.

The performance of the ML detector is easy to determine for perfect codes. With independent noise components, the probability of  $m$  bit errors in a block of  $n$  bits has a binomial distribution,

$$P(m, n) = \binom{n}{m} p^m (1-p)^{n-m}, \quad \binom{n}{m} = \frac{n!}{m!(n-m)!}. \quad (13.28)$$

For perfect codes, a block decoding error is sure to occur if more than  $t$  bit errors occur, so

$$\Pr[\text{block error}] = \sum_{m=t+1}^n P(m, n) = 1 - \sum_{m=0}^t P(m, n). \quad (13.29)$$

**Example 13-17.**

The (7,4) Hamming code is perfect, so

$$\Pr[\text{block error}] = 1 - (1-p)^7 - 7p(1-p)^6. \quad (13.30)$$

We can now compare the hard and soft decoders for the (7,4) Hamming code. If  $p = 0.01$ , then  $\Pr[\text{block error}] \approx 0.002$ , so coding appears to reduce the probability of error by a factor of 5. This result is deceptive, however, since on a white noise channel the coded system requires 7/4 times the bandwidth of the uncoded system, and hence

$$\sigma_c^2 = \frac{7}{4} \sigma_u^2. \quad (13.31)$$

The coded system thus has a larger BSC error probability  $p$  than the uncoded system, if the information rate is held constant. An exact comparison (Exercise 13-2) is tedious.  $\square$

**Exercise 13-2.**

- (a) Find upper and lower bounds on the probability of bit error of the (7,4) Hamming code in terms of  $\Pr[\text{block error}]$ .
- (b) Show that to achieve a probability of bit error of  $10^{-5}$ , the coded system uses a fraction of one dB less power than the uncoded system; i.e., the coding gain of the coded system is only a fraction of a dB. Assume that for both the coded and uncoded system the line code is binary antipodal.
- (c) Repeat (b) for a probability of bit error of  $10^{-7}$ .  $\square$

The result of this exercise is disappointing, since the gain is so small. This binary block code is a more reasonable choice on a channel where noise power is not such a strong function of symbol rate.

It is possible to use information theory (the principles of Chapter 4) to determine that with theoretical codes achieving error-free transmission, a soft decoder is between 1 and 2 dB better than a hard decoder (see for example [6], pages 270-275). The 1 dB difference occurs for codes with rate near one, and the 2 dB difference occurs for codes with rate near zero. The (7,4) Hamming code with a soft decoder has a power advantage on the order of 1-3 dB, according to Example 13-12, which suggests a 1-2 dB improvement over hard decoding.

The usefulness of (13.29) is limited to perfect codes. For those codes that are not perfect, (13.29) is an upper bound,

$$\Pr[\text{block error}] \leq \sum_{m=t+1}^n P(m,n), \quad (13.32)$$

because *some* error patterns with more than  $t$  bits errors will be corrected by the ML decoder. This bound often gives a good estimate. Many practical codes are *quasiperfect*, meaning that although some error patterns with  $t+1$  bit errors are corrected, none with  $t+2$  or more are corrected. For these we can get a lower bound

$$\Pr[\text{block error}] \geq \sum_{m=t+2}^n P(m,n). \quad (13.33)$$

Together these upper and lower bounds lead to good estimates of the performance of codes that are sometimes easier to use than the bounds of Section 9.2.

Many other bounds, both tighter and looser, are known. We refer the interested reader to the extensive coding literature.

**13.1.3. Parity-Check Matrix**

ML soft and hard decoders find the codeword closest (in Euclidean or Hamming distance) to the received block. Direct implementation becomes difficult for large  $k$  and  $n$ , since there are  $2^k$  distances that need to be computed and compared. Fortunately, for hard decoding, efficient techniques have evolved. The basic approach is to design the code to have a rich *algebraic* structure, and then exploit that structure in the decoding process. Recently, algebraic techniques have also been applied to soft decoding. Although we cannot give a comprehensive treatment of decoding



techniques, we can at least illustrate some of the most important concepts.

Consider a systematic linear  $(n, k)$  binary block code, which has a generator matrix of the form

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]. \quad (13.34)$$

Given a row vector  $\mathbf{b}$  of  $k$  bits, the corresponding codeword is  $\mathbf{c} = \mathbf{bG}$ , a row vector which can be written

$$\mathbf{c} = [\mathbf{b} \quad \mathbf{a}] \quad (13.35)$$

where  $\mathbf{a} = \mathbf{bP}$  is a row vector with  $n - k$  parity-check bits. Note that since  $\mathbf{a} = \mathbf{bP}$ ,

$$\mathbf{bP} \oplus \mathbf{a} = \mathbf{0} \quad (13.36)$$

where the modulo-two addition is performed element-wise. This can be written

$$[\mathbf{b} \quad \mathbf{a}] \begin{bmatrix} \mathbf{P} \\ \mathbf{I}_{n-k} \end{bmatrix} = \mathbf{0} \quad (13.37)$$

or

$$\mathbf{cH}' = \mathbf{0} \quad (13.38)$$

where

$$\mathbf{H} = [\mathbf{P}' \mid \mathbf{I}_{n-k}]. \quad (13.39)$$

$\mathbf{H}$  is called a *parity-check matrix*, because it can be used to test to see if a vector  $\mathbf{c}$  is a codeword by checking (13.38).

#### Example 13-18.

The parity-check matrix for the  $(k+1, k)$  single-parity-check code of Example 13-5 and Example 13-6 is

$$\mathbf{H} = [1 \ 1 \ \cdots \ 1 \ 1]. \quad (13.40)$$

As we already knew, we can check a bit vector to see if it is a codeword by summing (modulo-two) all of the bits and checking to see if the sum is zero.  $\square$

#### Example 13-19.

The parity-check matrix for the  $(7,4)$  Hamming code of Example 13-7 is

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (13.41)$$

An example of a codeword is  $\mathbf{c}_1 = [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1]$ , which satisfies  $\mathbf{c}_1 \mathbf{H}' = \mathbf{0}$ . By contrast,  $\mathbf{c}_2 = [0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1]$  is not a codeword, since  $\mathbf{c}_2 \mathbf{H}' = [1 \ 0 \ 0]$ .  $\square$

Although we have shown only how to get a parity-check matrix given the generator matrix of a systematic code, it is possible to find one for any linear block code. In fact, the parity-check matrix can be a compact and useful representation of the code.

For a received vector  $\tilde{c}$ , calculation of the distance to every codeword can be avoided by using a parity-check matrix. Write

$$\tilde{c} = c \oplus e \quad (13.42)$$

where  $c$  is the transmitted codeword and  $e$  is the error pattern. Define the *syndrome* to be

$$s = \tilde{c}H' = cH' + eH' = eH'. \quad (13.43)$$

Thus the syndrome depends only on the error pattern  $e$  and not on the transmitted codeword. The syndrome is zero if and only if  $e$  is a codeword (recall that  $0$  is always a codeword of a linear code). Efficient decoders use the syndrome to represent the error pattern, which can then be corrected.

### 13.1.4. Hamming Codes

Quite a variety of block codes have been developed, each with its advantages and disadvantages. We will describe a very small subset, beginning with Hamming codes.

The (7,4) Hamming code has figured prominently throughout this section. For any positive integer  $m$  there exists a Hamming code with

$$(n, k) = (2^m - 1, 2^m - 1 - m). \quad (13.44)$$

Recall that a parity-check matrix for an  $(n, k)$  code has  $n$  columns, each with  $n - k$  bits. Any systematic linear code is completely characterized by this matrix. For a Hamming code, the parity-check matrix is constructed by letting the  $n = 2^m - 1$  columns be all possible binary vectors with  $m = n - k$  elements, except the zero vector.

#### Example 13-20.

We have already seen the parity-check matrix for the (7,4) Hamming code:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (13.45)$$

The columns consist of all possible nonzero three-bit vectors, here arranged in systematic form.  $\square$

Once  $H$  is found, the generator matrix  $G$  for a systematic code can be found by comparing (13.39) and (13.34).

#### Exercise 13-3.

Show that every Hamming code has  $d_{H, \min} = 3$ . Consequently, all Hamming codes can correct single errors ( $t = 1$ ).  $\square$

Hamming codes are perfect codes, meaning that every length  $n$  bit pattern has distance 0 or 1 from exactly one codeword.

### 13.1.5. Cyclic Codes

Many practical block codes are *cyclic codes*, which have rich algebraic properties that lead to efficient decoding techniques. Some of the flavor of this algebra can be obtained from the discussion of Galois fields in appendices 10-A and 11-A. See Section 13.3 for some further reading on this topic.

An  $(n, k)$  linear block code is said to be *cyclic* if any cyclic shift of a codeword produces another codeword. All Hamming codes can be put into cyclic form.

#### Exercise 13-4.

Verify that all cyclic shifts of 1000101, namely

1100010 0110001 1011000 0101100 0010110 0001011 1000101

are codewords of the (7,4) Hamming code.  $\square$

The algebraic properties of cyclic codes permit collapsing the information contained by the generator matrix into a single polynomial, not surprisingly called the *generator polynomial*. Manipulations of this polynomial representation are powerful, permitting the synthesis of good codes and efficient coding and decoding techniques.

### 13.1.6. BCH and Reed-Solomon Codes

*BCH codes*, named after the inventors, Bose, Ray-Chaudhuri, and Hocquenghem, are a large class of multiple-error-correcting codes invented around 1960. For any positive integers  $m$  and  $t$ , there is a  $t$ -error-correcting binary BCH code with

$$n = 2^m - 1 \quad k \geq n - mt. \quad (13.46)$$

In order to correct  $t$  errors, it is clear that the minimum Hamming distance is bounded by

$$d_{H, \min} \geq 2t + 1. \quad (13.47)$$

BCH codes are important primarily because practical and efficient decoding techniques have been found [7], and because of the flexibility in the choice of parameters ( $n$  and  $k$ ).

An important class of nonbinary BCH codes are *Reed-Solomon codes*, in which the symbols are blocks of bits. Their importance is again the existence of practical decoding techniques, as well as their ability to correct bursts of errors.

### 13.1.7. Maximal-Length Shift Register Codes

In order to give a taste of cyclic codes without getting involved with the algebraic techniques that are required for a general treatment, consider a class of codes called *maximal-length shift register codes*. They are practically much less important than the BCH and Reed-Solomon codes, but can be described without introducing any new techniques. Maximal-length shift registers are described in Appendix 12-A.

**Example 13-21.**

A maximal-length feedback shift register with  $m = 4$  stages is shown in Figure 13-6.  $\square$

A block coder using a circuit such as that in Figure 13-6 operates as follows: source bits are divided into blocks of length  $m$ . The shift register is loaded with these bits and clocked  $2^m - 1$  times. The output from the circuit is then regarded as a codeword of length  $2^m - 1$ . The result is an  $(n, k) = (2^m - 1, m)$  block code. By picking the appropriate output from the circuit, the code is easily made systematic.

**Example 13-22.**

A systematic linear  $(15, 4)$  block code is generated by the system illustrated in Figure 13-6 if the output codeword is  $C_k = X_{k-4}$ . The first four bits out are the source bits.  $\square$

The rate of maximal-length shift register codes is

$$k/n = m/(2^m - 1) \quad (13.48)$$

which becomes very small as  $m$  becomes large. This limits the usefulness of these codes.

**Exercise 13-5.**

Show that a systematic maximal-length shift register code is a parity check code, and hence is linear.  $\square$

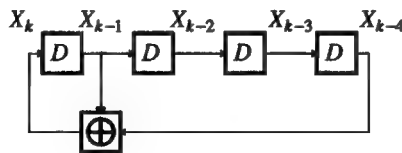
**Exercise 13-6.**

Show that the Hamming weight of all nonzero codewords of a maximal-length shift register code is  $2^{m-1}$ , where  $m$  is the length of the shift register.  $\square$

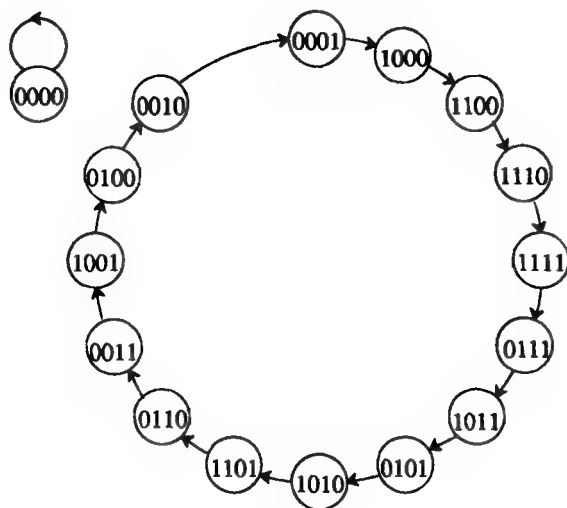
**Example 13-23.**

In the  $(15, 4)$  maximal-length shift register code, all nonzero codewords have Hamming weight 8. Since the code is linear, the minimum Hamming distance is 8, and the code can correct up to 3 bit errors.  $\square$

Maximal-length shift register codes are cyclic codes. This is easily seen by examining a state transition diagram of a maximal-length shift register.



**Figure 13-6.** A feedback shift register with  $m = 4$  stages. If the shift register is loaded with an initial 4-bit pattern, then as the shift register is clocked, the output will be a periodic bit sequence with period  $2^m - 1 = 15$ . This example is a maximal-length feedback shift register.



**Figure 13-7.** A state transition diagram for the the circuit in Figure 13-6. From its circular structure we see that the block code that it generates is cyclic.

#### Example 13-24.

A state transition diagram for the  $m = 4$  maximal-length shift register is shown in Example 13-24. From its circular structure we see that the initial condition determines where in the circle to start. Consequently, every nonzero codeword is a cyclic shift of every other nonzero codeword.  $\square$

## 13.2. CONVOLUTIONAL CODES

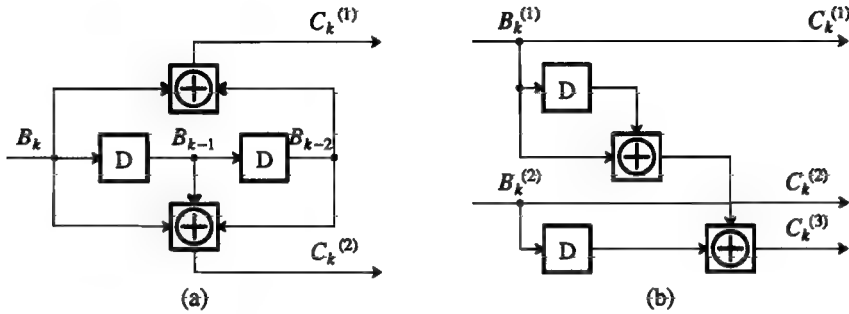
A *convolutional coder* is a finite memory system (rather than a memoryless system, as in the case of the block coder). The name refers to the fact that the added redundant bits are generated by modulo-two convolutions.

#### Example 13-25.

Two convolutional coder examples are shown in Figure 13-8. We will use these example coders often in this section.  $\square$

Convolutional codes are more commonly used than block codes, primarily because they are conceptually and practically simpler, and their performance matches (and often exceeds) that of good block codes. Their good performance is attributable in part to the availability of practical soft decoding techniques. Convolutional codes become particularly simple to describe once block codes are understood.

Just as for block codes, linear convolutional coders are constructed using modulo-two adders with the addition of delay elements. A convolutional coder can be



**Figure 13-8.** Two convolutional coders. a. A rate-1/2 convolutional coder denoted  $\text{conv}(1/2)$ . b. A rate-2/3 convolutional coder denoted  $\text{conv}(2/3)$ . The coder in (b) is systematic because the source bits appear directly in the coded bits.

described using a *generator matrix* where instead of the entries being zero or one, the entries are polynomials in  $D$  with coefficients that are either zero or one. As in Chapter 12, we adhere to the convention of using  $D$  in place of  $z^{-1}$  as the unit delay. These polynomials are modulo-two  $D$  transforms, defined by (12.66).

**Example 13-26.**

The generator matrix for the  $\text{conv}(1/2)$  coder of Figure 13-8a is

$$G(D) = [1 \oplus D^2, 1 \oplus D \oplus D^2], \quad (13.49)$$

and the generator matrix for  $\text{conv}(2/3)$  in Figure 13-8b is

$$G(D) = \begin{bmatrix} 1 & 0 & 1 \oplus D \\ 0 & 1 & D \end{bmatrix}. \quad (13.50)$$

□

The entries in the generator matrix are transfer functions. Define the row vectors

$$\mathbf{B}(D) = [B^{(1)}(D), \dots, B^{(k)}(D)] \quad (13.51)$$

$$C(D) = [C^{(1)}(D), \dots, C^{(n)}(D)] \quad (13.52)$$

where  $B^{(i)}(D)$  and  $C^{(i)}(D)$  are modulo-two  $D$  transforms of  $B_k^{(i)}$  and  $C_k^{(i)}$  (see Figure 13-8). The convolutional coder is defined by the matrix  $D$ -transform relation

$$C(D) = \mathbf{B}(D)G(D) \quad (13.53)$$

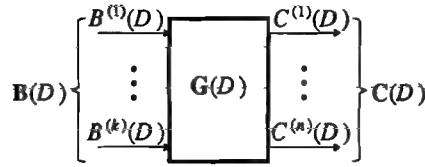
The notation for a general convolutional coder is summarized in Figure 13-9. Just as with block codes, a convolutional code has a *parity-check matrix*.

**Example 13-27.**

Comparing (13.50) with (13.34) and (13.39), a parity-check matrix for  $\text{conv}(2/3)$  is

$$\mathbf{H}(D) = [1 \oplus D, D, 1]. \quad (13.54)$$

□



**Figure 13-9.** A general convolutional coder. The modulo-two D-transform input and output sequences are shown, as is the generator matrix.

**Exercise 13-7.**

Verify that for  $\text{conv}(2/3)$  and  $H(D)$  given by (13.54),

$$C(D)H'(D) = 0 \quad (13.55)$$

for all code sequences  $C(D)$ .  $\square$

Just as with block codes, the parity-check matrix is a compact specification of the code. If the code is systematic, the generator matrix is easy to derive from the parity-check matrix, or vice versa. If the code is not systematic, then some modulo-two algebra may be required.

**Example 13-28.**

From (13.49) and (13.53) we know that

$$C^{(1)}(D) = (1 \oplus D^2)B(D) \quad (13.56)$$

$$C^{(2)}(D) = (1 \oplus D \oplus D^2)B(D)$$

Multiply both sides of the first equation by  $(1 \oplus D \oplus D^2)$  and of the second equation by  $(1 \oplus D^2)$  and notice that the two right hand sides are equal. Hence the left hand sides are equal,

$$C^{(1)}(D)(1 \oplus D \oplus D^2) = C^{(2)}(D)(1 \oplus D^2) \quad (13.57)$$

or

$$C^{(1)}(D)(1 \oplus D \oplus D^2) \oplus C^{(2)}(D)(1 \oplus D^2) = 0. \quad (13.58)$$

Hence a parity-check matrix is

$$H(D) = [1 \oplus D \oplus D^2, 1 \oplus D^2]. \quad (13.59)$$

$\square$

The *constraint length* of a convolutional code may be defined as one plus the maximum degree of the polynomials in the generator matrix (the maximum length of any impulse response):

$$M = 1 + \max_{i,j} [\deg(g_{ij}(D))]. \quad (13.60)$$

The  $\text{conv}(1/2)$  coder in Figure 13-8a has  $M = 3$ , while  $\text{conv}(2/3)$  in Figure 13-8b has  $M = 2$ . (This definition of constraint length is common but not universal in the literature.)

Given a parity-check matrix, it is often easy to design a systematic encoder.

**Example 13-29.**

Given

$$H(D) = [1 \oplus D, D, 1], \quad (13.61)$$

we will now derive the encoder shown in Figure 13-8b. From (13.55) we know that

$$(1 \oplus D)C^{(1)}(D) \oplus DC^{(2)}(D) \oplus C^{(3)}(D) = 0. \quad (13.62)$$

To get a systematic code, assign

$$C^{(1)}(D) = B^{(1)}(D) \quad \text{and} \quad C^{(2)}(D) = B^{(2)}(D). \quad (13.63)$$

Then note from (13.62) that

$$C^{(3)}(D) = (1 \oplus D)C^{(1)}(D) \oplus DC^{(2)}(D) \quad (13.64)$$

or

$$C^{(3)}(D) = (1 \oplus D)B^{(1)}(D) \oplus DB^{(2)}(D). \quad (13.65)$$

In the time domain this is

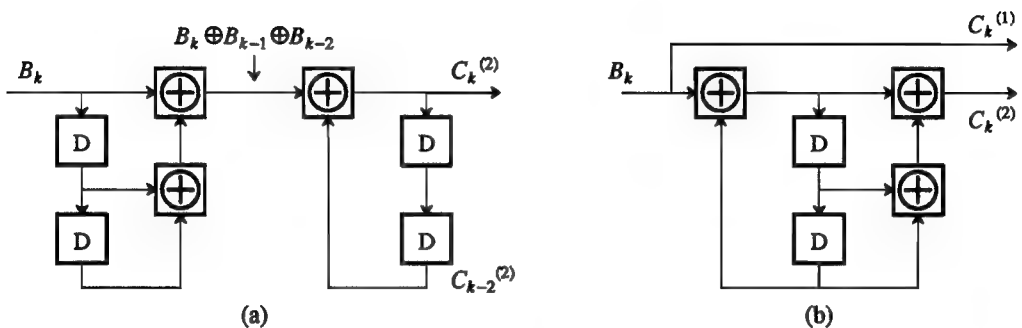
$$C_k^{(3)} = B_k^{(1)} \oplus B_{k-1}^{(1)} \oplus B_{k-1}^{(2)} \quad (13.66)$$

which is implemented in Figure 13-8b.  $\square$

Given a parity-check matrix, it is not always quite so simple to find a systematic implementation.

**Example 13-30.**

Consider the parity-check matrix for **conv(1/2)** given in (13.59). The encoder given in Figure 13-8a is not systematic, but there is an encoder that is systematic and has the same parity-check matrix. In all important respects the resulting code will be equivalent. From



**Figure 13-10.** Two versions of a systematic coder that has the same parity-check matrix as **conv(1/2)**. The one on the right is obtained by inverting the order of the two stages in the one on the left and combining the two delay lines into one.



(13.55),

$$(1 \oplus D \oplus D^2)C^{(1)}(D) \oplus (1 \oplus D^2)C^{(2)}(D) = 0. \quad (13.67)$$

Again, to get a systematic code, set  $C^{(1)}(D) = B(D)$ . From (13.67),

$$(1 \oplus D \oplus D^2)B(D) = (1 \oplus D^2)C^{(2)}(D). \quad (13.68)$$

Given this equation we can construct the system shown in Figure 13-10a that generates  $C_k^{(2)}$  from  $B_k$ . It is a cascade of two linear subsystems, the order of which can be reversed to get the implementation in Figure 13-10b. Those readers familiar with the design of recursive digital filters will recognize this procedure.  $\square$

For every  $k$  input bits,  $n$  bits are produced by the coder, so the *rate* of the convolutional coder is  $R = k/n$ . As with block codes, the rate is defined as the ratio of the input bit rate to output bit rate. In the performance calculations to follow, assume that the output bit stream is transmitted by a binary line code, requiring an increase in symbol rate due to the coding. This may not be appropriate for many bandlimited channels, where increasing the size of the alphabet may be more advantageous (the resulting codes are called *trellis codes* or *signal-space codes*, and are discussed in Chapter 14). Convolutional coding is preferred if the bandwidth is readily available, or if a large alphabet cannot be tolerated.

#### Example 13-31.

On optical fiber, it generally is more advantageous to retain binary signaling and increase the bit rate. On radio channels, nonlinearities are a significant problem, and binary signals are less susceptible to these nonlinearities. On channels with jamming, where spread spectrum might be used, bandwidth is not a consideration and in-band noise is not increased by increasing the bandwidth. These are examples where convolutional coders may be preferable to trellis coders.  $\square$

Like block codes, convolutional codes can either be used to *correct* errors or to *prevent* errors, depending on whether hard or soft decoding is used. We will study both types, again emphasizing soft decoding because it yields better performance and is closely tied into demodulation.

The representation of convolutional codes using generator matrices highlights their similarity to block codes. However, to do ML detection of the coded sequence, it is more convenient to represent convolutional codes as Markov chains. It is clear from Figure 13-8 that a convolutional code is a *shift register process* as defined in Figure 9-14. If the input sequence  $B_k$  is i.i.d., then the output  $C_k$  of the coder is a Markov chain, and we can define the state of the Markov chain to be the bits stored in the delay elements in the coder.

#### Example 13-32.

For conv(1/2), define the state of the Markov chain to be

$$\Psi_k = [B_{k-1}, B_{k-2}]. \quad (13.69)$$

The state transition diagram is shown in Figure 9-16, and the trellis diagram is shown in Figure 13-11. Recall from Section 9.6 that the trellis illustrates the progression through states over time. The transitions in the trellis in Figure 13-11b are labeled with the

(input,output) pairs  $(B_k, [C_k^{(1)}, C_k^{(2)}])$ . In Figure 13-11c they are labeled with the transmitted symbols instead of the coded bits, assuming binary antipodal signaling.  $\square$

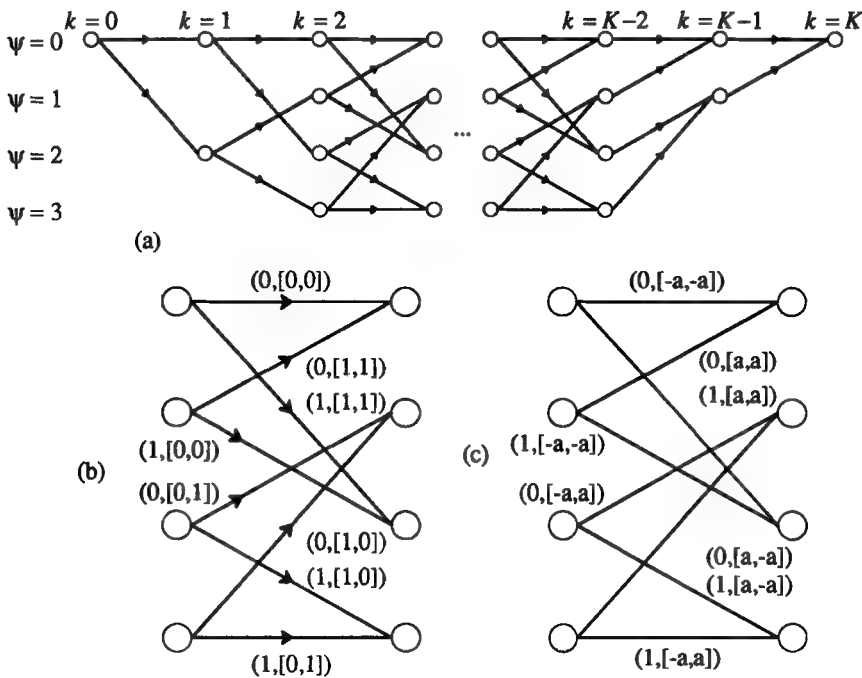
**Example 13-33.**

For  $\text{conv}(2/3)$  the state is

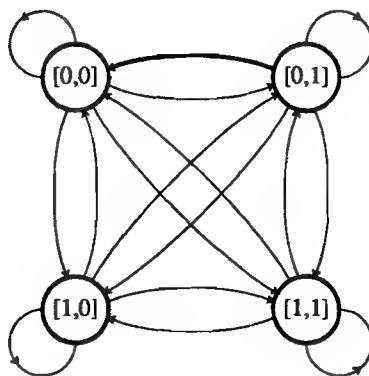
$$\Psi_k = [B_{k-1}^{(1)}, B_{k-1}^{(2)}]. \quad (13.70)$$

The state transition diagram for this code is fully connected, as shown in Figure 13-12.  $\square$

If the noise generation model has independent noise components, as we usually assume, then we can use the Viterbi algorithm (Section 9.6) for ML detection of the coded signal. In the soft decoding case with Gaussian noise a Euclidean distance branch metric is appropriate, while in the hard decoding case with a BSC it is Hamming distance.



**Figure 13-11.** a. A four-state trellis illustrating all possible state transitions of the Markov chain in Figure 9-16, which represents  $\text{conv}(1/2)$ , assuming the initial and termination states are zero. b. One stage of the trellis is shown with the transitions labeled with  $(B_k, [C_k^{(1)}, C_k^{(2)}])$ , where  $B_k$  is the input bit that triggers the transition and  $C_k^{(1)}$  and  $C_k^{(2)}$  are the outputs produced. c. The branches of the trellis labeled with the transmitted symbols rather than the bits out of the coder for binary antipodal signaling with levels  $\pm a$ .



**Figure 13-12.** A state transition diagram for a Markov chain modeling the convolutional coder of Figure 13-8b.

### 13.2.1. Performance of Soft Decoders

A coded transmission system with a soft decoder is shown in Figure 13-13. Assume additive Gaussian noise with independent noise components, and binary antipodal signaling with alphabet  $\Omega_A = \{\pm a\}$ . The signal level is  $a_u$  ( $a_c$ ) and the noise variance is  $\sigma_u^2$  ( $\sigma_c^2$ ) for the uncoded (coded) case.

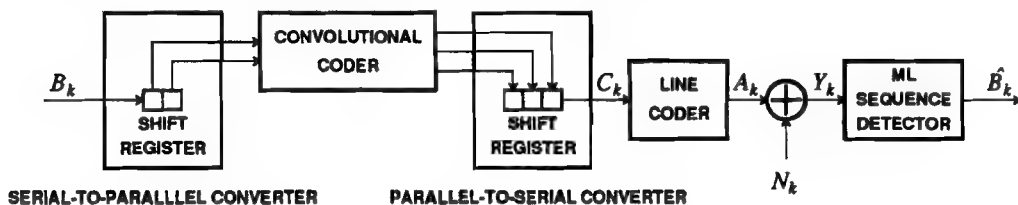
#### Example 13-34.

For conv(1/2) of Figure 13-8a, the coded system transmits at twice the symbol rate, occupying twice the bandwidth (assuming the same pulse shape is used). If a white Gaussian noise channel is used, the receiver will admit twice the noise power, and

$$\sigma_c = \sqrt{2}\sigma_u. \quad (13.71)$$

We will see that the coding gain more than makes up for the increased noise due to the bandwidth increase. Otherwise the coding scheme would not be useful!  $\square$

Again we can compute the coding gain, or reduction in SNR at the detector in the coded case for the same error probability, and then determine the difference in signal



**Figure 13-13.** A coded transmission system with a soft decoder (the ML sequence detector).

level required for the same continuous-time channel. The former does not take into account the different bandwidth of the two systems, while the latter does.

For the uncoded system, the probability of symbol error is equal to the probability of a bit error,

$$\Pr[\text{bit error}] = Q(a_u/\sigma_u). \quad (13.72)$$

For the coded system with soft decoder, instead of labeling trellis branches with output bits, they should be labeled with output *symbols*.

**Example 13-35.**

For  $\text{conv}(1/2)$  of Figure 13-8a, the trellis is shown in Figure 13-11c. Every transition is triggered by an input bit and produces a pair of output symbols. Given the corresponding pair of noisy observations  $y_k$  and  $y_{k+1}$ , and for a branch corresponding to symbols  $[d_k, d_{k+1}]$ , the ML detector computes the branch metric

$$d^2 = (y_k - d_k)^2 + (y_{k+1} - d_{k+1})^2. \quad (13.73)$$

The path metric, the sum of branch metrics, is the square of the Euclidean distance between the observation and the transmitted symbols for that path.  $\square$

The ML detector for a soft decoder selects the path through the trellis with the minimum path metric, as shown in Section 9.6. From Appendix 9-C, the probability of symbol error at high SNR is

$$\Pr[\text{symbol error}] \approx KQ(d_{\min}/2\sigma) \quad (13.74)$$

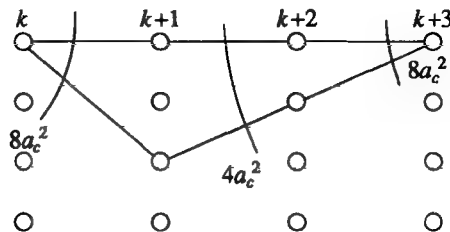
where  $d_{\min}$  is the minimum Euclidean distance of an error event and  $K$  is a constant between  $P$  and  $R$  given by (9.158) and (9.150).

**Example 13-36.**

For the trellis of Figure 13-11c, the error event with the minimum distance is shown in Figure 13-14 when the correct path is the all-zero sequence. The square of the Euclidean distance between the correct path and the error event is

$$d_{\min}^2 = 20a_c^2. \quad (13.75)$$

Every possible correct path through the trellis has exactly one minimum distance error



**Figure 13-14.** Minimum distance error event for  $\text{conv}(1/2)$ .

event, and each such error event  $e$  has one symbol error,  $w(e) = 1$ , so  $P = K = R = 1$ , and

$$\Pr[\text{symbol error}] \approx Q\left[\frac{\sqrt{20}a_c}{2\sigma_c}\right] = Q\left[\sqrt{5}\frac{a_c}{\sigma_c}\right]. \quad (13.76)$$

Each symbol represents one bit, so

$$\Pr[\text{bit error}] = \Pr[\text{symbol error}]. \quad (13.77)$$

Setting the arguments of  $Q(\cdot)$  equal for the coded and uncoded case, we see that  $SNR_u = \sqrt{5}SNR_c$  so the distance gain is  $10\log_{10}5 = 7$  dB. On a white Gaussian noise channel, 3 dB of this gain is lost because of the larger bandwidth in the coded case, so the net coding gain of the coded system is 4 dB. This is a significant improvement over the uncoded system, and also is considerably better than the specific block codes considered in Section 13.1. The approximations are valid when the probability of error is low.  $\square$

In Example 13-36, finding  $K$  was easy because  $P$  and  $R$  were both equal to one. In general, however, finding  $P$  and  $R$  is more difficult. Furthermore, finding  $d_{\min}$  can be tedious. The general technique described in Section 9.6, which uses the Viterbi algorithm to find  $d_{\min}$ , works for all cases. If the code is linear (Appendix 13-A), then the task is greatly simplified because only one actual path through the trellis must be considered. In this case, either the Viterbi algorithm technique of Section 9.6 or the signal flow graph technique of Appendix 13-B can be used.

### 13.2.2. Performance of Hard Decoders

Next we compare the hard decoder to both the soft decoder and the uncoded system. For the hard decoder, the channel and receiver front end can again be modeled as a BSC. In this case the appropriate branch metric is the Hamming distance between the received bits and the transmitted bits corresponding to that branch. Each branch has a set of  $L$  output bits associated with it ( $L = 2$  in  $\text{conv}(1/2)$  and  $L = 3$  in  $\text{conv}(2/3)$ ). The Hamming distance branch metric is an integer between 0 and  $L$ . The path metric is the sum of these branch metrics. We can use the Viterbi algorithm to implement the ML detector, which chooses the path through the trellis with minimum path metric. The analysis is similar to the Gaussian noise case, but  $Q(\cdot)$  is replaced by  $Q(\cdot, \cdot)$  given by (9.25). Suppose the codeword  $\mathbf{c}$  is transmitted and the channel error probability is  $p$ . Then the probability that the received bits are closer in Hamming distance to another codeword  $\hat{\mathbf{c}}$  is  $Q(d, p)$ , where  $d$  is the Hamming distance between  $\mathbf{c}$  and  $\hat{\mathbf{c}}$ .

#### Example 13-37.

For the coder  $\text{conv}(1/2)$  in Figure 13-8a, the minimum-distance error event has length  $K = 2$  and is the same as for the soft decoder shown in Figure 13-14. We will bound the probability that this particular error event begins at some time  $k = i$ . Write the noise-free input codeword as

$$\mathbf{c} = [\mathbf{c}_i, \mathbf{c}_{i+1}, \mathbf{c}_{i+2}] \quad (13.78)$$

where each  $\mathbf{c}_k$  is a pair of bits determined by a state transition. (We can consider codewords of finite length only because we are considering an error event of finite length.) Assume a zero state trajectory,  $\psi_k = 0$  for all  $k$ , so  $\mathbf{c} = [(0,0), (0,0), (0,0)]$ . The minimum distance error event in Figure 13-14 has a corresponding codeword  $\hat{\mathbf{c}} = [(1,1), (0,1), (1,1)]$

and is Hamming distance five from  $\mathbf{c}$ . This is exactly the situation described in Example 9-18! The observation  $\hat{\mathbf{c}}$  will be closer to  $\hat{\mathbf{c}}$  than to  $\mathbf{c}$  if three or more bits are changed in the five positions in which the two codewords differ. The probability of this occurring is given by (9.26), so

$$\Pr[\text{this error event}] \leq Q(5p) = 10p^3(1-p)^2 + 5p^4(1-p) + p^5. \quad (13.79)$$

This is the same bound given in (9.147).  $\square$

Appendix 9-C shows that when  $p$  is small, the probability of a detection error is approximately  $KQ(d_{\min}, p)$  (see (9.162)) for some constant  $K$  between  $P$  and  $R$  given by (9.158) and (9.150). As in the Gaussian case, the situation is simple if every possible actual path through the trellis has exactly one minimum distance error event, and that minimum distance error event has exactly one detection error. In this case,  $P = R = K = 1$ .

#### Example 13-38.

The minimum distance error event in Example 13-37 has exactly one detection error (the first detected bit of the three erroneous stages will be incorrect, see Figure 13-11b). Furthermore, since the code is linear, every actual path through the trellis also has exactly one minimum distance error event with exactly one detection error. From (9.162) we can assert

$$\Pr[\text{bit error}] = \Pr[\text{detection error}] \approx 10p^3(1-p)^2 + 5p^4(1-p) + p^5. \quad (13.80)$$

It is assumed that other error events are far less likely than the minimum distance error event. Compare (13.80) to the probability of bit error  $p$  of an uncoded system. If  $p = 0.1$ , then the uncoded system has a probability of bit error of 0.1, while the coded system has probability of bit error approximately 0.0086.  $\square$

In general, evaluating  $Q(d_{\min}, p)$  exactly can be tedious. Fortunately, for  $p$  close to zero, the first term in the summation in (9.25) will dominate, so

$$Q(d_{\min}, p) \approx \binom{d_{\min}}{t+1} p^{t+1} (1-p)^{d-t+1}, \quad (13.81)$$

where  $t$  is given by (9.24).

#### Example 13-39.

Continuing Example 13-38, using (13.81) we get

$$\Pr[\text{bit error}] \approx 10p^3 \quad (13.82)$$

as long as  $p$  is close to zero.  $\square$

In order to compare hard and soft decoders, we must relate the error probability of the BSC to the SNR on the channel prior to the slicer.

#### Example 13-40.

To make a rough approximation, we can determine the symbol amplitudes  $a_u$ ,  $a_h$  and  $a_s$  required to achieve a *particular* probability of error, say  $10^{-5}$ . Solving

$$10p^3 = 10^{-5} \quad (13.83)$$

for  $p$  we find  $p \approx 0.01$ . Hence  $p$  is close to zero and our approximation is valid for this probability of error. To achieve  $p \approx 0.01$  on an additive Gaussian white noise channel with

binary antipodal signaling it is necessary that

$$Q(a_h/\sigma_c) \approx 0.01 \quad (13.84)$$

where  $\pm a_h$  are the binary antipodal levels and  $\sigma_c$  is the noise variance of the coded system. In terms of the noise of the uncoded system,

$$Q(a_h/\sqrt{2}\sigma_u) \approx 0.01. \quad (13.85)$$

This is satisfied by  $a_h \approx 3.4\sigma_u$ . (This number can be found using the same approximation techniques as in Problem 13-1.) For the uncoded system, the probability of bit error is equal to approximately  $10^{-5}$  when  $a_u = 4.3\sigma_u$ . This is roughly  $20\log(4.3/3.4) \approx 2.0$  dB more power for the uncoded system. So the coding gain of the hard decoded system is of the order of 2 dB, far short of the 4 dB obtained by the soft decoded system.  $\square$

Although our analysis has been limited to simple examples, we conclude that just as with block codes, hard decoders for convolutional codes yield less coding gain than do soft decoders. As with block codes, this does not mean that hard decoders are not used. Their implementation may be simpler, they can improve existing transmission systems with minimal modification, and the gain on channels with other than Gaussian white noise may be better.

### 13.3. HISTORICAL NOTES AND FURTHER READING

In 1948, Shannon started the field of coding theory by demonstrating that through coding it is theoretically possible to achieve error-free transmission on noisy communication channels (Chapter 4). His results did not bound decoding complexity and delay, but nevertheless suggested that improvements in error probability could be achieved using practical codes. The first block codes, introduced in 1950 by Hamming, were capable of correcting single errors in hard-decoding system, but fell disappointingly short of the capacity predicted by Shannon. The next major advance emerged only after another 10 years, with the BCH codes in 1959 and the closely-related Reed-Solomon codes in 1960. The strong algebraic properties of these codes led to efforts to find efficient coding and decoding algorithms. Convolutional codes appeared in 1957. Initially, they lacked the algebraic properties relating to distance in block codes, but this shortcoming has been partially remedied. During the 1970s, these avenues of research continued, but the next major breakthrough did not occur until 1982 with the description of trellis codes (Chapter 14) by Ungerboeck.

A classic coding theory book, concentrating on block codes with algebraic (hard) decoding, is that of Berlekamp [8], who will convince any reader of the beauty of the subject. A more recent comprehensive text is Blahut [9]. A standard textbook that is still used, 25 years after its publication, is Gallager [10], which includes an extensive discussion of the performance of various block codes. A comprehensive treatment of block codes is MacWilliams and Sloane [11]. A particularly useful paper on convolutional codes is Massey [12]. McEliece [13] gives a very readable treatment of both information theory and coding. The chapter on convolutional codes is an excellent introduction to the subject. More detailed information about convolutional codes can

be obtained from Forney [14], where equivalent realizations of codes are discussed. The second edition of a classic book by Peterson, written with Weldon, discusses efficient encoding and decoding of cyclic codes [7]. A more advanced text with a comprehensive treatment of convolutional codes is Viterbi and Omura [15]. Tables of good convolutional codes can be found in [16,17].

## APPENDIX 13-A LINEARITY OF CODES

In order to treat linearity of codes in a general way, it is helpful to review some definitions from algebra. The linearity of codes turns out to be important enough to justify this digression.

A *ring*  $R$  is a set with two operations defined; we will call these operations *addition* and *multiplication* although they may bear little resemblance to ordinary addition and multiplication. To emphasize that they may be different, we denote addition by  $\oplus$  and multiplication by  $*$ .

The most important feature of the addition operation is that adding any two operands in  $R$  produces a result in  $R$ .

### Example 13-41.

The set of real numbers  $\mathbf{R}$  is a ring, and addition is ordinary. The sum of any two real numbers is a real number. The set of binary digits  $Z_2 = \{0,1\}$  is a ring, where addition is modulo-two.  $\square$

As with ordinary addition,  $\oplus$  must be associative

$$(r_1 \oplus r_2) \oplus r_3 = r_1 \oplus (r_2 \oplus r_3) \quad (13.86)$$

and commutative

$$r_1 \oplus r_2 = r_2 \oplus r_1. \quad (13.87)$$

Furthermore, the ring must have an *additive identity*, denoted "0", such that  $r \oplus 0 = r$  for any  $r$  in  $R$ . Finally, every element  $r$  must have an *additive inverse* which when added to  $r$  produces the zero element.

### Example 13-42.

The additive identity in  $\mathbf{R}$  is zero, and the additive inverse of any  $r \in \mathbf{R}$  is  $-r$ . The additive identity in  $Z_2$  is zero, and the additive inverse of  $z \in Z_2$  is  $z$  itself.  $\square$

The multiplication operation  $*$  is similarly defined to operate on two elements in the ring to produce an element in the ring. Like ordinary multiplication it must be associative, but need not be commutative. It must however be distributive over addition,

$$r_1 * (r_2 \oplus r_3) = r_1 * r_2 \oplus r_1 * r_3 \quad (r_1 \oplus r_2) * r_3 = r_1 * r_3 \oplus r_2 * r_3. \quad (13.88)$$



To summarize, a ring has (1) closure under  $\oplus$  and  $*$ , (2) associativity for  $\oplus$  and  $*$ , (3) distributivity of  $*$  over  $\oplus$  (13.88), (4) commutativity of  $\oplus$ , (5) an additive identity 0, and (6) an additive inverse  $-r$ .

**Exercise 13-8.**

Show from the above properties that  $r*0=0$  for all  $r \in R$ .  $\square$

A *field* is a ring where

- There is a *multiplicative identity* in the ring, denoted 1, such that  $r*1 = r$ ,
- Multiplication is commutative ( $r_1*r_2 = r_2*r_1$ ), and
- There is a *multiplicative inverse*  $1/r \in R$  for every element of the field except the additive identity 0. The multiplicative inverse satisfies  $r*(1/r) = 1$ .

**Example 13-43.**

Both  $\mathbf{R}$  and  $Z_2$  from Example 13-41 are fields, but the set of positive integers  $Z_{\infty}$  is not because there is not a multiplicative inverse for all elements of the field. The set of all integers (positive and negative) is also not a field, but unlike  $Z_{\infty}$  it is a ring.  $\square$

A field with a finite number of elements is called a *finite field*, or a *Galois field*. It turns out that all finite fields have  $p^m$  elements, where  $p$  is a prime number and  $m$  is any integer.

**Example 13-44.**

There is no field with six elements, but there is a field with three elements. Let  $Z_3 = \{0,1,2\}$ ,  $\oplus$  be modulo three addition, and  $*$  be modulo three multiplication. It is easy to verify that all of the above properties of fields apply, with the possible exception of the multiplicative inverse. We can verify that the multiplicative inverse exists for all elements from the following table:

element	inverse
0	none
1	1
2	2

$\square$

**Exercise 13-9.**

Verify that  $Z_5 = \{0,1,2,3,4\}$  is a field with addition and multiplication modulo 5. List the multiplicative inverses.  $\square$

For both  $Z_3$  and  $Z_5$ , the number of elements is prime. If the number of elements is not prime, but is  $q = p^m$  for  $m > 1$  and  $p$  prime, then multiplication in the field is more complicated than simple modulo- $q$  multiplication.

**Exercise 13-10.**

Verify that multiplication in the field  $Z_4 = \{0,1,2,3\}$  is not simple modulo-four multiplication. **Hint:** Try to find the multiplicative inverse of each of the elements. Construct a multiplication table that satisfies the requirements for multiplication in a field.  $\square$

Fortunately, in this book we are primarily interested in  $Z_2$ , so we need not get distracted by these complications. Suffice it to say that it is possible to define addition and multiplication so that  $GF(q)$  is a field for any  $q = p^m$ ,  $p$  prime.

All finite fields with  $q$  elements are equivalent and are denoted  $GF(q)$ . Hence any two-element field is equivalent to  $Z_2$ , or  $GF(2)$ .

A vector space  $V_n(GF(q))$  over the field  $GF(q)$  is a set of  $n$ -tuples of elements from the field. These are much like the vector spaces of Section 2.6, the only difference being that they are defined over Galois fields rather than the fields of real or complex numbers. Addition ( $\oplus$ ) of two vectors is defined element-wise, and must produce a vector in the vector space (in other words the vector space is *closed under vector addition*). Vector addition is commutative and associative, and there is an additive identity (the zero vector) and an additive inverse (the negative of a vector). A vector can be multiplied element-wise by a scalar in the field, and the vector space is closed under scalar multiplication. Scalar multiplication is associative and distributive, from the properties of multiplication in the field.

Using these definitions, we first study the linearity of block codes, and then turn to convolutional codes.

**Block Codes**

An  $(n, k)$  block code is a set of  $2^k$  vectors of length  $n$  in  $V_n(GF(2))$ . The *Hamming distance*  $d_H(c_1, c_2)$  between  $c_1$  and  $c_2$  in  $V_n(GF(2))$  is the number of differing bits between  $c_1$  and  $c_2$ . The *Hamming weight*  $w_H(c)$  of  $c$  in  $V_n(GF(2))$  is the number of ones in  $c$ . Clearly

$$d_H(c_1, c_2) = w_H(c_1 \oplus c_2) \quad (13.89)$$

because  $c_1 \oplus c_2$  has a component equal to one only in positions where  $c_1$  and  $c_2$  differ. To determine the performance of a code we are interested in finding

$$d_{H,min} = \min_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}} d_H(c_1, c_2) \quad (13.90)$$

which from (13.89) is

$$d_{H,min} = \min_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}} w_H(c_1 \oplus c_2). \quad (13.91)$$

To find this minimum distance it appears that we have to search over all pairs of code-words  $c_1$  and  $c_2$ . Fortunately for linear codes this is not necessary.

An  $(n, k)$  linear block code  $C$  is a  $k$ -dimensional subspace of  $V_n(GF(2))$ . By *subspace* we mean that  $C$  itself is a vector space, and hence is closed under vector

addition. I.e., if  $\mathbf{c}_1$  and  $\mathbf{c}_2$  are in  $C$ , then  $\mathbf{c}_1 \oplus \mathbf{c}_2 \in C$ . Hence (13.91) becomes

$$d_{H,min} = \min_{\substack{\mathbf{c} \in C \\ \mathbf{c} \neq \mathbf{0}}} w_H(\mathbf{c}). \quad (13.92)$$

To find the minimum Hamming distance in a linear code we need only find the minimum Hamming weight in the linear code. Equivalently, set  $\mathbf{c}_1 = \mathbf{0}$  and let  $\mathbf{c}_2$  vary over all codewords in (13.91). In other words, when trying to find the probability of the most likely decoding error, assume that the zero vector is transmitted and then consider the probability of decoding to a non-zero codeword.

A *basis* of a vector space is a set of vectors such that every element in the vector space can be expressed as a linear combination of the vectors in the basis. Since the vector space is closed under scalar multiplication and vector addition, every linear combination of basis vectors must be in the vector space. We can use this fact to show that every linear block code can be generated using a generator matrix as shown in (13.3). Simply let the rows of the generator matrix  $\mathbf{G}$  be a set of basis vectors for the code. Then any codeword may be written  $\mathbf{B}\mathbf{G}$ , where  $\mathbf{B}$  is a binary vector of length  $k$ . It can also be shown that any linear block code has a systematic generator matrix (simply permute the columns of  $\mathbf{G}$ ).

Associated with any  $(n, k)$  linear code  $C$  is a *dual code*  $C_D$ . The dual code is an  $(n, n - k)$  linear code consisting of the set of vectors orthogonal to all vectors in  $C$ . (The vector  $\mathbf{x}$  is orthogonal to  $\mathbf{c}$  if  $\mathbf{x}\mathbf{c}' = 0$ , where  $\mathbf{c}'$  is the transpose of  $\mathbf{c}$ .) If  $\mathbf{G}$  is the generator matrix and  $\mathbf{H}$  is a parity-check matrix for  $C$ , then  $\mathbf{H}$  is a generator matrix for  $C_D$ , and  $\mathbf{G}$  is a parity-check matrix.

#### Exercise 13-11.

Show that the rows of  $\mathbf{H}$  are orthogonal to all codewords in  $C$ . Then show that  $\mathbf{G}$  is a parity-check matrix for the code generated by  $\mathbf{H}$ .  $\square$

## Convolutional Codes

To study the linearity of convolutional codes we use the definition from (13.53), reproduced here,

$$\mathbf{C}(D) = \mathbf{B}(D)\mathbf{G}(D) \quad (13.93)$$

where  $\mathbf{B}(D)$  is a  $k$ -tuple of polynomials in  $D$ ,  $\mathbf{C}(D)$  is an  $n$ -tuple of polynomials in  $D$ , and  $\mathbf{G}(D)$  is a  $k \times n$  matrix of polynomials in  $D$ . Each polynomial has coefficients in  $GF(2)$ .

#### Exercise 13-12.

Let  $F_D$  be the set of polynomials in  $D$  with coefficients in  $GF(2)$ . Verify that  $F_D$  is a ring.  $\square$

If we augment  $F_D$  to include rational polynomials in  $D$ , then it is a field as well. Define a vector space  $V_n(F_D)$  of  $n$ -tuples of polynomials in  $F_D$ . The codewords  $\mathbf{C}(D)$  are in  $V_n(F_D)$ . Furthermore, from (13.93),  $\mathbf{C}(D)$  is formed by linear combinations of the rows of  $\mathbf{G}(D)$ . The rows of  $\mathbf{G}(D)$ , which are also in  $V_n(F_D)$ , form a basis

for the code. The code is therefore a subspace and hence linear.

To find  $d_{H,min}$  for the code, define the Hamming distance  $d_H(C_1(D), C_2(D))$  between two codewords to be the total number of differing coefficients in the polynomials  $C_1(D)$  and  $C_2(D)$ . The Hamming weight  $w_H(C(D))$  of  $C(D)$  is the total number of non-zero coefficients in  $C(D)$ . Then just as with block codes,

$$d_H(C_1(D), C_2(D)) = w_H(C_1(D) \oplus C_2(D)). \quad (13.94)$$

Since the code is linear,  $C_1(D) \oplus C_2(D)$  is a codeword and

$$d_{H,min} = \min_{\substack{C(D) \in C \\ C(D) \neq 0}} w_H(C(D)). \quad (13.95)$$

We conclude that linear convolutional codes behave like linear block codes in that the minimum Hamming distance is the minimum Hamming weight. Put another way, we can safely assume the transmitted sequence is all zero and find the probability of the code sequence that is closest in Hamming distance to the zero sequence.

### Linearity in Signal Space

From the above results we can find the minimum Hamming distance for linear block and convolutional codes with relative ease. The performance of hard decoders is determined primarily by this distance. However, the performance of soft decoders and signal space codes (e.g. trellis codes, covered in Chapter 14) is determined by the minimum *Euclidean* distance between coded *symbol* sequences, rather than *Hamming* distance between *bit* sequences. Often, when considering symbol sequences, linearity does not apply, so we *cannot* safely assume that the zero codeword is transmitted and find the minimum distance from it. An important exception is when the line coder is binary, meaning that the alphabet has only two symbols. Assume the codeword  $\mathbf{c}_i$  (a binary vector) with corresponding symbols  $\mathbf{a}_i$  is transmitted. Assume binary signaling, with symbols selected from the set  $\Omega_A = \{a, b\}$ . The Euclidean distance between  $\mathbf{a}_i$  and  $\mathbf{a}_j \in \Omega_A$  is

$$d_E(\mathbf{a}_i, \mathbf{a}_j) = |a - b| \sqrt{d_H(\mathbf{c}_i, \mathbf{c}_j)} \quad (13.96)$$

where  $d_H(\mathbf{c}_i, \mathbf{c}_j)$  is the Hamming distance.

#### Example 13-45.

Two codewords of a (3,2) simple parity-check code are  $\mathbf{c}_1 = [1,0,1]$  and  $\mathbf{c}_2 = [1,1,0]$ . Then  $d_H(\mathbf{c}_1, \mathbf{c}_2) = 2$ . If binary antipodal signaling with  $\Omega_A = \{\pm 1\}$  is used, then  $\mathbf{a}_1 = [+1, -1, +1]$  and  $\mathbf{a}_2 = [+1, +1, -1]$ . The Euclidean distance is  $d_E(\mathbf{a}_1, \mathbf{a}_2) = 2\sqrt{2}$ , in agreement with (13.96).

□

Hence the minimum Euclidean distance between a transmitted sequence  $\mathbf{a}_i$  and any other sequence  $\mathbf{a}_j$  is

$$d_{E,min} = |a - b| \sqrt{d_{H,min}} \quad (13.97)$$

From the above arguments, if the code is linear,  $d_{H,min}$  can be found by assuming the zero codeword is being transmitted and finding the Hamming distance of all other codewords from the zero codeword.

When the signaling is not binary, things are not so simple. It is still not necessary to consider all possible pairs of codewords, however, in order to find the minimum Euclidean distance between pairs of codewords. Some straightforward simplifications are possible. Again, let  $\mathbf{a}_i$  and  $\mathbf{a}_j$  be two blocks of symbols corresponding to two codewords. Then

$$d_E(\mathbf{a}_i, \mathbf{a}_j) = w_E(\mathbf{e}) \quad (13.98)$$

where  $w_E(\mathbf{e})$  is the *Euclidean weight* of the vector  $\mathbf{e} = \mathbf{a}_i - \mathbf{a}_j$  (the square root of the sum of the squares of the elements). Now to find  $d_{E,min}$ , observe

$$d_{E,min} = \min_{\mathbf{e} \in \Omega_e} w_E(\mathbf{e}). \quad (13.99)$$

It is important to note, however, that  $\Omega_e$  is not the set of permitted symbol sequences.

#### Example 13-46.

Continuing the previous example, note that

$$w_E(\mathbf{a}_1 - \mathbf{a}_2) = w_E([0, -2, +2]) = 2\sqrt{2} \quad (13.100)$$

in agreement with previous results. The vector  $\mathbf{e} = [0, -2, +2]$  does not consist of symbols.  $\square$

The set  $\Omega_e$  is generally significantly smaller than the set of all possible pairs of codewords, so the search space for  $d_{E,min}$  is reduced. This is essentially the same technique used in Section 9.6 in the determination of the minimum distance of ISI sequences! The complexity can be reduced further by observing that  $w_E(\mathbf{e}) = w_E(|\mathbf{e}|)$  and searching over  $\Omega_{|\mathbf{e}|}$ .

#### Example 13-47.

Continuing Example 13-46,  $\Omega_{|\mathbf{e}|}$  is

$$\Omega_{|\mathbf{e}|} = \{[0, 2, 2], [2, 0, 2], [2, 2, 0]\}, \quad (13.101)$$

so again  $d_{E,min} = 2\sqrt{2}$ . A similar reduction of the search space is used in Section 9.6 to find the minimum distance between transmitted signals in an ISI channel.

## APPENDIX 13-B PATH ENUMERATORS

To estimate the performance of convolutional codes we need to find the most probable error event. For simple examples it can be found by inspection, but this method is seriously prone to error. In Section 9.6 we described a general technique that requires assuming an actual state trajectory and then using the Viterbi algorithm to find the minimum-distance error event for that state trajectory. In general, all possible actual state trajectories must be considered. In Section 9.6 we showed that this is not necessary for ISI examples, because we can exploit the linearity of the signal generation model. Fortunately, it is also not necessary for linear codes, which turn out to

be even simpler than the ISI case. As shown in Appendix 13-A, for linear codes it is sufficient to consider only one actual path through the trellis. Hence, we can use the Viterbi algorithm to find the minimum-distance error event for any assumed actual path through the trellis, as detailed in Section 9.6, and the distance will be the global minimum distance. As indicated in the Appendix 13-A, this will work for linear codes using a hard decoder, or a soft decoder when the line code is binary. In this appendix, we give an alternative technique using signal flow graphs (as in Section 3.3). Although this is not necessarily simpler than using the Viterbi algorithm, the technique can be used to simultaneously compute essentially all the information about the error events, such as their length, the number of bit errors in each one, and the number of error events at each distance. It gives much more information than just the minimum distance.

The general technique requires an assumption that the correct state trajectory remains in the zero state. This is why the technique is restricted to linear codes! An error event is therefore a path that leaves the zero state and later returns. We can enumerate all such paths, as shown in the following example.

#### Example 13-48.

Consider the convolutional coder in Figure 13-20 (this coder is studied in Problem 13-11). The state transition diagram is shown in Figure 13-15. The branches in the diagram are labeled with the variable  $z$  raised to a power equal to the Hamming distance of that branch from the zero branch. For instance, the label  $z^0 = 1$  follows from the fact that this branch is the zero branch, and hence its distance is zero. We are interested in the distance of all paths from the zero state back to the zero state. To find it, break the zero state into two states as shown in Figure 13-16. If we view this graph as a signal flow graph, then the gain from the  $O_1$  state to the  $O_2$  state is a polynomial in  $z$  that enumerates the weights of all possible paths from the zero state back to itself. By inspection that polynomial is

$$T(z) = z^3 + z^4 + z^5 + \cdots \quad (13.102)$$

It is easy to see that error event with the minimum Hamming distance has Hamming distance 3, the error event with the second smallest Hamming distance has distance 4, etc. The polynomial  $T(z)$  is called a path-enumerator polynomial. It can be expressed more compactly as

$$T(z) = \frac{z^3}{1 - z} . \quad (13.103)$$

It can be readily verified that this is the same as (13.102) by carrying out the long division. The same technique can be used to enumerate the Euclidean distances of error events in

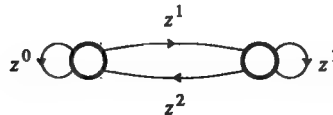


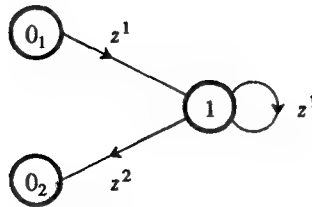
Figure 13-15. The state transition diagram for the convolutional coder in Problem 13-11.

order to determine the performance of a soft decoder as long as the alphabet is binary (see Problem 13-14).  $\square$

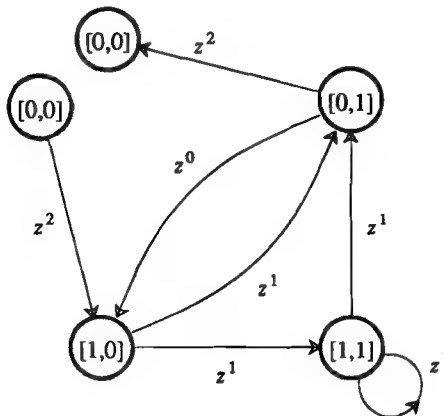
The convolutional coder considered above does not have nearly as much coding gain as others of comparable complexity that we have considered. Unfortunately, for most useful linear codes, constructing the path enumerator polynomial by inspection can be difficult. Fortunately, a computer program or well-documented techniques from the literature can be used (for example, Mason's gain formula [18,19,20] is useful).

#### Example 13-49.

The convolutional coder state transition diagram of Figure 9-16 is modified as shown in Figure 13-17. The branch weights are again the variable  $z$  raised to the power of the Hamming distance of the branch from the zero branch. It is possible to show that



**Figure 13-16.** To enumerate the paths from the zero state back to itself, break the zero state in two as shown. The weights on the arcs are a variable  $z$  raised to the power of the Hamming distance from the zero path.



**Figure 13-17.** The state transition diagram of Figure 9-16 is modified for the purpose of enumerating the paths from the  $[0,0]$  state back to itself. The branch weights are equal to  $z$  raised to the Hamming weight of the branch.

$$T(z) = \frac{z^5}{1-2z} \quad (13.104)$$

By long division

$$T(z) = z^5 + 2z^6 + 4z^7 + 8z^8 + \dots \quad (13.105)$$

This says that there is one error event with Hamming distance 5 (which we knew already), two error events with Hamming distance 6, four with distance 7, etc.  $\square$

The path enumerator technique can be used to obtain a variety of information about a code in addition to the distances of the error events. The extension is simple. In Problem 13-15 we show how to enumerate the number of bit errors in the error events and the length of the error events.

## PROBLEMS

- 13-1. Verify the numbers in the table in Example 13-11. **Hint:** Use one of the approximations for  $Q(\cdot)$  in Figure 3-1.
- 13-2. An  $(n, 1)$  *repetition code* is one where each bit is repeated  $n$  times.
- Compute the minimum Hamming distance  $d_{H,min}$  of such a code as a function of  $n$ .
  - How does this coding technique compare with the (7,4) Hamming code with a hard decoder?
- 13-3. Consider a linear block code  $C$  with parity-check matrix  $\mathbf{H}$  and minimum Hamming distance  $d_{H,min}$  between codewords.
- Show that  $d_{H,min}$  is equal to the minimum number of columns of  $\mathbf{H}$  that can be added to produce 0.
  - Use part (a) to show that for all linear block codes

$$d_{H,min} \leq n - k + 1. \quad (13.106)$$

- 13-4. Given the generator matrix for a (7,3) linear block code

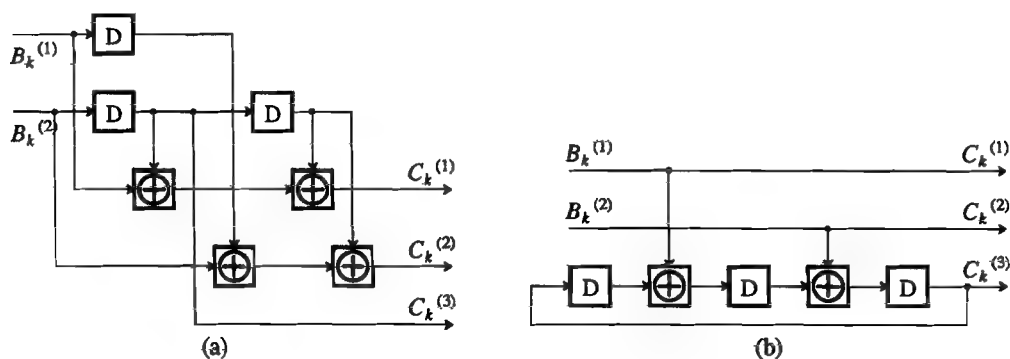
$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \quad (13.107)$$

- Construct the generator matrix of an equivalent systematic code.
  - Find the parity-check matrix  $\mathbf{H}$ .
  - Construct a table of all possible syndromes  $\mathbf{s}$  and find the error pattern  $\mathbf{e}$  most likely to have resulted in that syndrome.
  - What is the relationship of this code to the (7,4) Hamming code?
  - Find  $d_{H,min}$ . How many bit errors in a block can be reliably corrected?
  - Find the codeword  $\mathbf{c} = \mathbf{bG}$  for  $\mathbf{b} = 101$  and verify that  $\mathbf{cH}^T = 0$ .
- 13-5.
- Give the generator and parity-check matrices in systematic form for the (15,11) Hamming code.
  - Find a parity-check matrix for a non-systematic (15,11) Hamming code such that the syndrome of any bit pattern with one bit error can be interpreted as a binary number that identifies

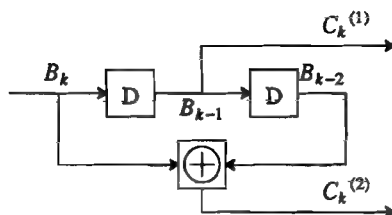


the position of the bit error.

- 13-6. Compare the performance of a (15,11) Hamming code with a soft decoder to an uncoded system with the same source bit rate. Assume that both the coded and uncoded systems use a binary antipodal line coder with alphabet  $\pm a$ . You may ignore the constant multiplying the  $Q(\cdot)$  function.
- 13-7. Estimate the power advantage of the (15,4) maximal-length shift register code with a soft decoder. You may ignore the constants in front of the  $Q(\cdot)$  term. To get the power advantage, compare to an uncoded system with the same line code (binary antipodal), and a sufficiently lower symbol rate that the source bit rate is the same. Assume additive white Gaussian noise on the channel.
- 13-8. Consider the non-systematic convolutional coder in Figure 13-18a.
- Find the parity-check matrix.
  - Show that Figure 13-18b has the same parity-check matrix.
- 13-9. Consider transmitting bits  $B_k$  over a BSC channel using the convolutional coder  $\text{conv}(1/2)$  of Figure 13-8a. Assume that  $B_k = 0$  for  $k < 0$  and  $k \geq K$ . Suppose  $K = 3$  and the observation sequence is  $\{0, 1, 0, 1, 1, 1, 0, 0, \dots\}$ . Draw the trellis for the Markov model and label the transition weights. What is the ML estimate of the incoming bit sequence?
- 13-10. Consider the rate-1/2 convolutional coder shown in Figure 13-19.



**Figure 13-18.** Two encoders with the same parity-check matrix, where (b) is systematic and (a) is not.



**Figure 13-19.** A rate-1/2 convolutional coder studied in Problem 13-10. It is not as good as the coder in Figure 13-8a.

- (a) Draw the state transition diagram and trellis with each transition labeled with  $(B_k, [C_k^{(1)}, C_k^{(2)}])$ .
- (b) Assume that  $C_k^{(1)}$  and  $C_k^{(2)}$  are interleaved on a BSC with probability  $p$  of flipping a bit. Find the error event with the minimum Hamming distance. Find the probability of the error event and compare with the probability computed in (13.79).
- (c) Assume that  $C_k^{(1)}$  and  $C_k^{(2)}$  are to be interleaved over an additive white Gaussian noise channel. Assume  $A_i$  is the symbol sequence chosen from the alphabet  $\Omega_A = \{-a, +a\}$ . Assume an ML soft decoder. Estimate the probability of error event and compare it to the uncoded system and to  $\text{conv}(1/2)$  with soft decoding whose performance is approximated in (13.76). Give the comparison in dB.

13-11. Consider the convolutional coder shown in Figure 13-20.

- (a) Find the Hamming distance of the minimum distance error event and give an upper bound (like that in (13.79)) for the probability of this error event using a hard decoder.
- (b) Assuming binary antipodal signaling with alphabet  $\pm a$ , find the error event with the minimum Euclidean distance. Estimate the coding gain using a soft decoder, assuming that this error event dominates. You may neglect the constant multiplying the  $\mathcal{Q}(\cdot)$  function.

13-12. Is the code consisting of the following codewords linear?

$$0010 \ 0100 \ 1110 \ 1000 \ 1010 \ 1100 \ 0110. \quad (13.108)$$

13-13. List the codewords of the dual of the (7,4) Hamming code. What are  $n$ ,  $k$ , and  $d_{H,\min}$ ?

13-14. Assuming binary antipodal signaling for the convolutional coder of Example 13-48, find an enumerator polynomial for the *squares* of the Euclidean distances.

13-15. In this problem we show how to use the path enumerator polynomials to find the distances, number of bit errors, and lengths of all error events simultaneously (for linear codes). Consider the convolutional coder in Figure 13-20. To find just the Hamming distances of the error events we use the broken state transition diagram of Figure 13-16. To find the number of bit errors and lengths of the error events we use the labels in Figure 13-21.

- (a) Find an expression for the gain from  $0_1$  to  $0_2$ . It will be a polynomial  $T(x, y, z)$  in  $x$ ,  $y$ , and  $z$ .
- (b) What is the distance and number of bit errors in the error event with length four (i.e. the error event that traverses five incorrect branches, or four incorrect states, before returning to the zero state)?
- (c) Suppose you are told that the gain from  $0_1$  to  $0_2$  in Figure 13-22 is

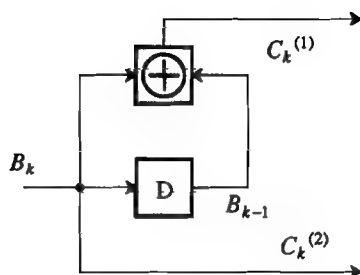
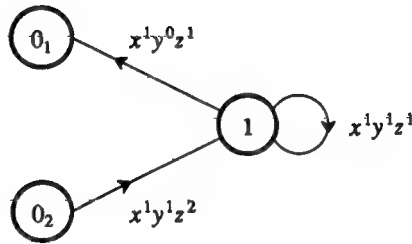
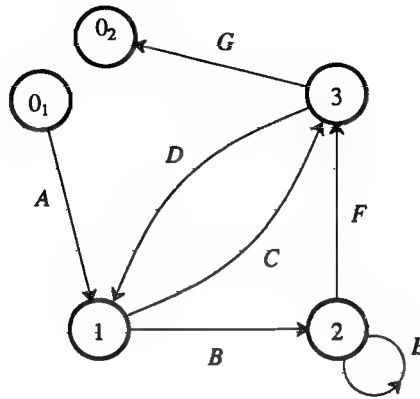


Figure 13-20. A convolutional coder studied in Problem 13-11 and Problem 13-15.



**Figure 13-21.** The state transition diagram of Figure 13-16 has been modified so that the exponent of  $x$  denotes the length of the branch (always one) and the exponent of  $y$  denotes the number of bit errors that occur if that branch is selected by the decoder (assuming the zero path is correct). The exponent of  $z$  still shows the Hamming weight of the branch.



**Figure 13-22.** A signal flow graph with branch weights shown.

$$T = \frac{ACG(1-E) + ABFG}{1 - DC - E - BFD + EDC} \quad (13.109)$$

For the convolutional coder of Example 13-49, determine the number of distance 6 error events and their lengths, and the number of length 4 error events and their distances.

## REFERENCES

1. D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. on Information Theory* IT-18 pp. 170-182 (Jan. 1972).
2. G. D. Forney, Jr., "Generalized Minimum Distance Decoding," *IEEE Trans. on Information Theory* IT-12 pp. 125-131 (April 1966).
3. S. Wainberg and J. K. Wolf, "Algebraic Decoding of Block Codes Over a  $q$ -ary Input,  $Q$ -ary Output Channel,  $Q > q$ ," *Information and Control* 22 pp. 232-247 (April 1973).
4. E. J. Weldon, Jr., "Decoding Binary Block Codes on  $Q$ -ary Output Channels," *IEEE Trans. on Information Theory* IT-17 pp. 713-718 (Nov. 1971).
5. J. K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Trans. on Information Theory* IT-24 pp. 76-81 (Jan. 1978).
6. J. G. Proakis, *Digital Communications, Second Edition*, McGraw-Hill Book Co., New York (1989).
7. W. Peterson and E. Weldon, *Error-Correcting Codes, 2nd Ed.*, M.I.T. Press, Cambridge, Mass (1972).
8. E. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill Book Co., New York (1968).
9. R. E. Blahut, "Theory and Practice of Error Control Codes," Addison-Wesley, (1983).
10. R. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, Inc., New York (1968).
11. F. J. MacWilliams and J. J. Sloane, *The Theory of Error Correcting Codes*, North-Holland, New York (1977).
12. J. Massey, "Error Bounds for Tree Codes, Trellis Codes, and Convolutional Codes with Encoding and Decoding Procedures," in *Coding and Complexity*, ed. G. Longo, Springer-Verlag, New York (1977).
13. R. J. McEliece, *The Theory of Information and Coding*, Addison Wesley Pub. Co. (1977).
14. G. D. Forney, Jr., "Convolutional Codes I: Algebraic Structure," *IEEE Trans. on Information Theory* IT-16 pp. 720-738 (Nov. 1970).
15. A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill (1979).
16. K. J. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rates  $1/2$ ,  $1/3$  and  $1/4$ ," *IEEE Trans. Information Theory* IT-19 pp. 371-372 (1973).
17. D. G. Daut, J. W. Modestino, and L. D. Wismer, "New Short Constraint Length Convolutional Code Construction for Selected Rational Rates," *IEEE Trans. Information Theory* IT-28 pp. 794-800 (1982).
18. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, Prentice-Hall, Englewood Cliffs, N.J. (1988).
19. S. J. Mason, "Feedback Theory — Further Properties of Signal Flow Graphs," *Proc. IRE* 44(7) p. 920 (July 1956).
20. B. C. Kuo, *Automatic Control Systems*, Prentice-Hall, Englewood Cliffs, N.J. (1962).

# 14

---

## SIGNAL-SPACE CODING

---

Error control, introduced in Chapter 13, adds redundancy, in the form of extra bits, and then uses that redundancy to correct errors introduced by the channel. There are shortcomings that we will address in this chapter:

- In typical applications, the extra bits either increase the bandwidth and the noise allowed into the receiver, or increase the number of bits per symbol. In the second case, the constellation minimum distance will be decreased or the average power increased.
- Convolutional and block codes specify how to generate redundant bits, but do not specify how to map the additional bits into data symbols.

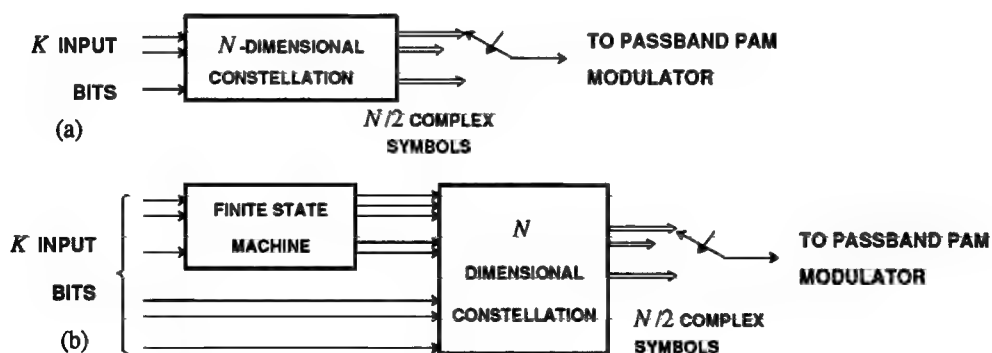
The best performance results if we use soft decoding (a Euclidean signal-space norm rather than Hamming-distance norm). The resulting error probability is accurately predicted by the signal-space minimum distance. That minimum distance is affected in turn by how we do the mapping of bits to symbols. In this chapter, we introduce a superior approach to designing codes for soft decoding. We directly consider the geometry of signal sets within signal space, with the goal of maximizing the minimum distance, while meeting power and bandwidth constraints. This does not mean that the algebraic techniques emphasized in Chapters 12 and 13 (groups, fields, etc.) must be abandoned, but rather that these algebraic tools should be used in conjunction with geometric considerations.

Codes designed by choosing signal sets in signal space with desirable geometric (as well as algebraic) properties are called *signal-space codes*. They are advantageous

on channels where spectral efficiency is at a premium (such as radio or voiceband telephone channels). To be practical, these signal-space codes must have a regular geometric and algebraic structure that can be used to simplify their implementation.

The key to obtaining large coding gains is to design codes in a subspace of signal space with high dimensionality, where a larger minimum distance in relation to signal power can be obtained. This is the essence of the channel coding theorem of Chapter 4. This high dimensionality does not necessarily imply a large bandwidth. For example, if we group together a large number of successive symbols in a PAM system, the resulting "vector" symbol is multidimensional. In other words, the dimensionality  $2Bt_0$  can be increased for fixed bandwidth  $B$  by increasing the time interval  $t_0$ , making it multiple symbol intervals.

Our starting point will be to consider the problem of designing a signal constellation (in  $N$ -dimensional Euclidean space) that has a large minimum distance in relation to its average power. This is identical to the problem of designing baseband or passband signal constellations considered in Chapter 6 for  $N = 1$  and  $N = 2$ , except that higher dimensionality is desired. We will show that it is advantageous to extend this design to  $N > 2$ , as illustrated in Figure 14-1a in the context of passband PAM modulation. A sequence of  $N/2$  two-dimensional (complex-valued) transmitted symbols can be considered as a single point in an  $N$  dimensional constellation. Each member of the constellation alphabet (called a codeword) is a vector in  $N$ -dimensional Euclidean space. Analogous to the coder in Chapter 6, but generalized to  $N$  dimensions, a set of  $K$  input bits are used to choose one of  $2^K$  codewords in the multidimensional constellation. That codeword is transmitted serially as  $N/2$  data symbols over a passband PAM modulation system.



**Figure 14-1.** Two basic ways of generating signal-space codes in conjunction with passband PAM modulation. (a) The signal constellation is generalized from two dimensions to  $N$  dimensions, corresponding to a vector of  $N/2$  transmitted symbols. (b) A finite state machine (FSM) is used to choose a vector from an  $N$ -dimensional constellation.

An example of a multidimensional constellation is a *lattice code*, which is a generalization of some of the rectangular constellations of Chapter 6, and has geometric and algebraic structure that makes it practical to synthesize, analyze, and implement. Although lattice codes can approach capacity for large  $N$ , the exponential increase in the number of codewords with  $N$  rules out their use for large  $N$ .

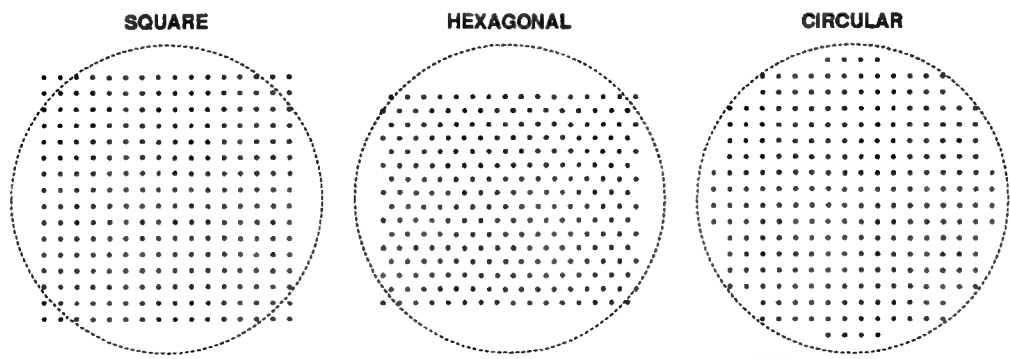
A way to achieve significant coding gain without the implementation complexity of lattice codes is to extend the dimensionality of the transmitted signal by basing it on a finite state machine (FSM). As shown in Figure 14-1b, a group of  $K$  input bits is divided into two groups. The first group drives an FSM, which introduces redundancy by generating more bits at its output than there are at its input. The FSM output, together with the second group of input bits, specifies one from among a set of codewords in an  $N$ -dimensional signal constellation. The extra bits produced by the FSM implies an inherent increase in the number of points in the constellation. As in the multidimensional constellation, the codewords are transmitted serially as  $N/2$  complex symbols. Not only are significant coding gains possible this way, but the implementation of the receiver maximum-likelihood detector (Chapter 9) can be based on the Viterbi algorithm, greatly reducing the complexity of soft decoding.

In Section 14.1 we will consider the design of multidimensional signal constellations, followed by trellis codes based on the FSM approach in Section 14.2. A generalization of the trellis code, the coset code, is introduced in Section 14.3. Finally, Section 14.4 discusses the combination of signal-space coding with ISI.

## 14.1. MULTIDIMENSIONAL SIGNAL CONSTELLATIONS

When passband PAM systems were designed in Chapters 6 through 10, a signal constellation was designed for a complex-valued data symbol, and this same constellation was used for each successive symbol. In Chapter 6, several intuitive design approaches for improving signal constellations were described, two of which are illustrated in Figure 14-2. Three constellations are shown in Figure 14-2 with the same minimum distance and 256 points. Thus, the three constellations will have the same spectral efficiency and, to accurate approximation, the same error probability. Where they differ is in the variance of the data symbols, which (assuming equally probable points) are clearly different. Using the square QAM constellation as a reference, the other two constellations illustrate two basic approaches to improving constellation design.

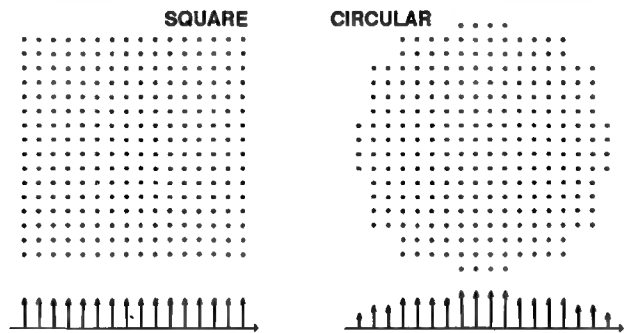
The first idea is to change the shape or outline of the constellation without changing the relative positioning of points (on a regular square grid). The "circular" constellation approximates a circular shape. This circular constellation will have a lower variance than the square constellation, because every point that is moved from outside the circle to the inside will make a smaller contribution to the variance as a result. On these same grounds, a circular shape will have the lowest variance of any shaping region for a square grid of points. The resulting reduction in signal power is called *shaping gain*.



**Figure 14-2.** A square QAM constellation, and two alternative constellations that illustrate shaping and coding gain. All three constellations have the same minimum distance, and each has 256 points.

Significantly, shaping the constellation changes the marginal density of the real part or imaginary part of the data symbol. This is illustrated in Figure 14-3, where the marginal density of the real-valued component of the complex symbol is compared for the square and circular constellations, assuming the points in the signal constellation are equally likely. For the circularly shaped constellation, the one-dimensional marginal density becomes nonuniform, even though the two-dimensional density *is* uniform.

A second approach to improving a constellation, also illustrated in Figure 14-2, is to change the relative spacing of points in the constellation. The hexagonal constellation, in which points fall on a grid of equilateral triangles, also reduces the variance for the same minimum distance. (Alternatively, we could keep the variance constant, in which case the hexagonal constellation would have a larger minimum distance than



**Figure 14-3.** The marginal probability density functions of one dimension for a an unshaped (square) and shaped (circular) two-dimensional constellation. The two-dimensional density is assumed to be uniform.



the square constellation.) This decrease in power for the same minimum distance or increase in minimum distance for the same power through changing the relative spacing of the points is called *coding gain*.

Coding and shaping gain can be combined, for example by changing the points in the circularly shaped constellation to a hexagonal grid while retaining the circular shaping.

If we define a constellation for a single data symbol  $a_k$ , as was done in Chapter 6, then it will be one-dimensional for the baseband case, and two-dimensional for the passband (complex) case. The square constellation of Figure 14-2 is simply the Cartesian product of a pair of identical one-dimensional constellations. However, introducing either shaping gain or coding gain, or both, implies the two-dimensional constellation is no longer the Cartesian product of one-dimensional constellations. The advantages of shaping and coding gain imply that it is preferable to design a two-dimensional constellation directly as an alphabet of points in two-dimensional space, rather than taking the "lazy" approach of forming a Cartesian product of one-dimensional constellations.

Neither shaping nor coding gain is feasible in one dimension, but both are available in two dimensions. If going from one to two dimensions is beneficial, could it be that moving to even higher dimensions is a good idea? In this section, we will introduce the third fundamental idea in constellation design, the *multidimensional signal constellation*. Taking the passband case, consider a sequence of complex-valued data symbols  $\{a_k, -\infty < k < \infty\}$ . A subset of  $N/2$  successive symbols  $\{a_k, a_{k+1}, \dots, a_{k+N/2-1}\}$  (where of course  $N$  is even), can be considered as a vector in  $N$ -dimensional real-valued Euclidean space. Our convention is that a data symbol drawn from this  $N$ -dimensional constellation is transmitted once every  $N/2$  symbol intervals. When we design a two-dimensional constellation, and choose the  $N/2$  successive symbols to be an arbitrary sequence of two-dimensional symbols drawn from that constellation, the resulting  $N$ -dimensional constellation is a Cartesian product of  $N/2$  two-dimensional constellations. An alternative is to design an  $N$ -dimensional constellation that is not constrained to have this Cartesian-product structure. This is then called an  *$N$ -dimensional signal constellation*. Whenever  $N > 2$ , it is called a multidimensional signal constellation.

Greater shaping and coding gains can be achieved with a multidimensional constellation than with a two-dimensional constellation. In Section 8.6 it was shown that two-dimensional constellations, on a Gaussian channel, suffer an "SNR gap to capacity". This gap can be closed completely with a multidimensional constellation as  $N \rightarrow \infty$ . This result is a straightforward application of the capacity theorem (Chapter 4) if the multidimensional constellation is not constrained, since it is simply a general channel code anticipated by the channel capacity theorem.. Significantly, as will be cited below, there exist constellations that have an imposed structure (that are a multidimensional generalization of the square and hexagonal constellations) that can also completely close the gap as  $N \rightarrow \infty$ .

Practically speaking, significant shaping and coding gains can be achieved for modest  $N$ . However, multidimensional constellations suffer from a complexity that increases exponentially with dimensionality. To mitigate this, multidimensional

constellations can be used in conjunction with trellis codes, as described in Section 14.3. Multidimensional constellations also serve to further our understanding of the structure of signal-space codes, and particularly the relationship between coding and shaping gain.

### 14.1.1. Lattice Codes

The two-dimensional constellations of Figure 14-2 are special cases of *lattice codes*. Let the dimension of the constellation be  $N$ , and let  $\{\mathbf{x}_1, \dots, \mathbf{x}_I\}$  be a set of  $I$  linearly independent basis vectors in  $N$ -dimensional Euclidean space. Of course, we must have that  $I \leq N$ . Consider the set of points in  $N$ -dimensional Euclidean space that can be expressed in the form

$$\mathbf{x} = \sum_{i=1}^I k_i \cdot \mathbf{x}_i, \quad (14.1)$$

where the  $\{k_1, \dots, k_I\}$  are integers. This countably infinite set of points is called a *lattice*, and is denoted by  $\Lambda$ . The regularly spaced set of points in a lattice is an appropriate choice for a multidimensional signal constellation for three reasons:

- Since the probability of error is determined by the minimum-distance, it is advantageous to choose regular arrays of points as in a lattice, where all points are equidistant from their neighbors.
- From an implementation perspective, the points in the constellation can be described and manipulated in terms of the vector of integers, using fixed-point integer arithmetic.
- The lattice has a convenient geometric and algebraic structure. Algebraically it is a *group*, meaning that it is closed with respect to vector summation and difference. One implication of this is that the zero vector must be a member of any lattice. This algebraic structure can be exploited in both implementation and in deriving various properties of the lattice.

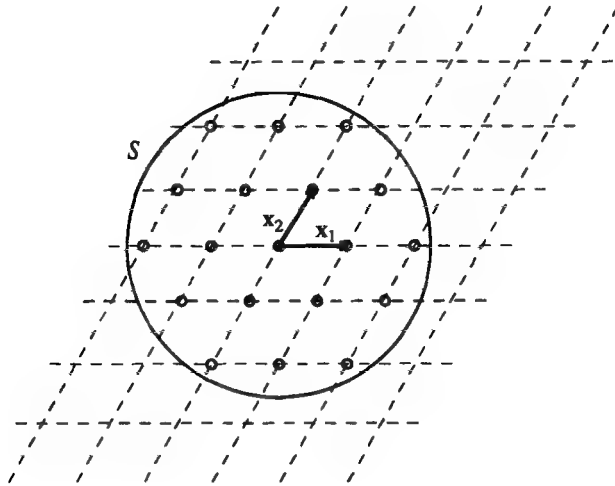
A *lattice code* is made up of three constituents. First, there is the lattice  $\Lambda$ . Second, there is a translation vector  $\mathbf{a}$ , and a translation of the lattice  $\Lambda + \mathbf{a}$  consisting of all points in the lattice translated by  $\mathbf{a}$ . The motivation for this translation is that a lattice is constrained to have a point at zero (corresponding to all-zero integers), and the translation frees us from the constraint, for example to minimize the transmitted power. Third, there is a finite region  $S$ , with the convention that the lattice code is the intersection of  $\Lambda$  with  $S$ . This results in a finite number of points in the lattice code, and also gives another degree of control over the transmitted power. To display all these parameters at once, we write the lattice code in the form  $(\Lambda + \mathbf{a}) \cap S$ .

#### Example 14-1.

The two-dimensional hexagonal constellation of Figure 14-2 can be described as a lattice code, as illustrated in Figure 14-4. The first step is to choose two basis vectors corresponding to the sides of an equilateral triangle,

$$\mathbf{x}_1 = [d, 0], \quad \mathbf{x}_2 = [d/2, \sqrt{3}d/2]. \quad (14.2)$$

In this case,  $d$  will be the minimum distance for the code. The second step is to choose



**Figure 14-4.** Illustration of a two-dimensional lattice code, where  $S$  is a circle (two-dimensional sphere). The lattice points fall at the intersection of the dotted lines.

some translation vector  $\mathbf{a}$  ( $\mathbf{a} = \mathbf{0}$  shown). The third step is to choose a region  $S$ , shown as a circle (two-dimensional sphere). Finally, the lattice code consists of all points on the lattice that fall within  $S$ .  $\square$

This example shows that many of the constellations illustrated in Chapter 6 (with the notable exception of the AM-PM constellations) can be formulated as lattice codes. The greatest significance of the lattice code formulation is that it readily extends to higher dimensions. Since it is difficult to draw or visualize dimensions higher than three, the mathematical structure of lattice codes becomes particularly important.

There are two properties of a lattice that influence its effectiveness as a code:

- The *minimum distance*  $d_{\min}(\Lambda)$  of points in the lattice relates directly to the error probability of the lattice code (for an additive Gaussian noise channel at high SNR).
- The *fundamental volume*  $V(\Lambda)$  is the volume of  $N$ -dimensional space associated with each lattice point. It is the inverse of the number of lattice points per unit volume. The fundamental volume is important because it relates directly to the number of lattice points within a given region  $S$ , and hence affects the spectral efficiency. (The spectral efficiency is affected by other factors as well, such as  $S$ , the PAM pulse bandwidth, etc.)

There is a direct relationship between the minimum distance and fundamental volume; increasing one tends to also increase the other. That is, for fixed transmit power, increasing the minimum distance tends to reduce the spectral efficiency.

### 14.1.2. Normalized SNR and Error Probability

For purposes of this chapter, we can assume that a symbol-rate discrete-time channel has been derived by demodulation and the sampled matched filter (Chapters 7 and 8). If the continuous-time channel has no ISI (the pulse autocorrelation  $\rho_h(k) = \delta_k$ ), then the noise samples at the matched filter output will be white and Gaussian. Thus, assume the equivalent discrete-time channel is

$$Y_k = X_k + Z_k, \quad (14.3)$$

where  $X_k$  and  $Y_k$  are complex-valued inputs and outputs and  $Z_k$  is circularly symmetric white Gaussian noise. As in Chapter 8, define  $\sigma^2$  as the variance per dimension, or  $E[|Z_k|^2] = 2\sigma^2$ . We will use discrete-time terminology, and call  $2\sigma^2$  the *noise power per two dimensions*. (In continuous time, the power is defined as the energy per unit time, while in discrete time we define it as the energy per sample.) Further, define  $P$  as the upper bound on the input *signal power per two dimensions* for the input to this discrete-time channel,

$$E[|X_k|^2] \leq P. \quad (14.4)$$

The *spectral efficiency*  $v$  is defined as the number of bits of information communicated per complex sample, or the *bits per two dimensions*.

#### Example 14-2.

A passband PAM system operating at the maximum symbol rate relative to the bandwidth of the underlying channel will have a spectral efficiency  $v$  bits/sec-Hz, since the symbol rate is equal to the bandwidth of the channel. Thus, for this idealized continuous-time channel, and only for this channel, the spectral efficiency as defined in (6.7) (in bits/sec-Hz), and the spectral efficiency of the discrete-time channel of (14.3) (in bits per two dimensions), will be numerically equal.  $\square$

The Shannon limit on spectral efficiency for the channel of (14.3),  $v_c$ , is given by (4.36) for  $N = 2$ ,

$$v_c = \log_2(1 + SNR), \quad SNR = P/2\sigma^2. \quad (14.5)$$

Equivalently, (14.5) can be written in the form

$$\frac{SNR}{2^{v_c} - 1} = 1. \quad (14.6)$$

As in Chapter 8, a rate-normalized signal-to-noise ratio can be defined for a system operating over channel (14.3) with signal-to-noise ratio  $SNR$  and spectral efficiency  $v$ ,

$$SNR_{\text{norm}} = \frac{SNR}{2^v - 1}. \quad (14.7)$$

$SNR_{\text{norm}}$  has the interpretation that the Shannon limit  $v \leq v_c$  is equivalent to  $SNR_{\text{norm}} > 1$ .

If complex symbols  $X_k = A_k$  are transmitted over (14.3), then from Chapter 8 the error probability of a minimum-distance receiver is accurately approximated by the

union bound as

$$P_e = K \cdot Q(d_{\min}/2\sigma), \quad (14.8)$$

where  $d_{\min}$  is the two-dimensional Euclidean minimum distance between known signals. If (14.3) is used to transmit a vector signal, then  $P_e$  is likewise given by (14.8), except  $d_{\min}$  is now the minimum Euclidean distance between vector signals, and the error coefficient  $K$  may be changed.

The error probability can be expressed in terms of the  $SNR_{\text{norm}}$ ,  $v$ , and  $P$ , by expressing the squared argument of  $Q(\cdot)$  as

$$\frac{d_{\min}^2}{4\sigma^2} = \frac{d_{\min}^2 (2^v - 1)}{2P} \cdot \frac{P}{2\sigma^2 (2^v - 1)} = \gamma \cdot SNR_{\text{norm}}, \quad (14.9)$$

where (14.5) and (14.7) have been used to define  $SNR_{\text{norm}}$ , and

$$\gamma = \frac{d_{\min}^2 (2^v - 1)}{2P}. \quad (14.10)$$

Given this definition,

$$P_e \approx K \cdot Q(\sqrt{\gamma \cdot SNR_{\text{norm}}}). \quad (14.11)$$

This expression for  $\gamma$  is similar to that derived in Chapter 8, except that it applies to vector signals,  $v$  is measured by bits per two dimensions, and  $P$  is power per two dimensions.

As shown in Figure 8-9, the parameter  $\gamma$  relates directly to the SNR gap to capacity, and in particular the larger  $\gamma$ , the lower the error probability and the smaller the SNR gap to capacity. It was also shown in Chapter 8 that  $\gamma=3$  for a square two-dimensional QAM constellation, independent of the constellation size.

### 14.1.3. Coding and Shaping Gains for Lattice Codes

We will now show that for lattice codes with a large number of points,  $\gamma$  can be approximated in a particularly simple and insightful way. This will allow us to extend the concepts of shaping and coding gains as illustrated in two dimensions in Figure 14-2 to multidimensional constellations.

#### The Continuous Approximation

The spectral efficiency and power of a constellation can be expressed in terms of the parameters of the lattice and the shaping region. Considering first the spectral efficiency, if the volume of an  $N$ -dimensional shaping region  $S$  is defined as  $V(S)$ , then the number of points in  $(\Lambda + \mathbf{a}) \cap S$  falling within  $S$  is accurately approximated by  $V(S)/V(\Lambda)$ , especially as this ratio gets large. Then for a given  $(\Lambda + \mathbf{a}) \cap S$ ,

$$v \approx \frac{2}{N} \log_2 \frac{V(S)}{V(\Lambda)}, \quad (14.12)$$

since the points are divided over  $N/2$  complex symbols.

To calculate the power for a constellation, the starting point is the probability density of the data symbols, which consists of delta functions at the constellation

points. When the number of points in  $(\Lambda + \mathbf{a}) \cap S$  is large, then the points in the lattice within region  $S$  are closely spaced at regular intervals, and their probability density can reasonably be approximated as continuous rather than discrete. When the signal points are equally likely, then the appropriate continuous density is uniform over the region  $S$ . In calculating the power of the constellation, a continuous uniform density is just the Riemann integral approximation to the sum that would correspond to the discrete density. Denote as  $P(S)$  the variance of a uniform distribution over  $S$ . Since the uniform distribution has height  $1/V(S)$ ,

$$P(S) = \frac{1}{V(S)} \int_S \|\mathbf{x}\|^2 d\mathbf{x}. \quad (14.13)$$

This is the *continuous approximation*, first invoked in Section 10.1.4 to study the properties of transmitter precoding.  $P(S)$  is an approximation to the power per  $N$  dimensions, and thus the continuous approximation for the power per two dimensions is

$$P \approx 2P(S)/N. \quad (14.14)$$

#### Example 14-3.

When  $N = 2$ , and the shaping region  $S$  is an  $2R \times 2R$  square,  $V(S) = (2R)^2$ , and

$$P \approx P(S) = \frac{1}{4R^2} \int_{-R}^R \int_{-R}^R (x_1^2 + x_2^2) dx_1 dx_2 = \frac{2R^2}{3}. \quad (14.15)$$

□

#### Example 14-4.

When  $N = 2$  and the shaping region  $S$  is a circle with radius  $R$ , then  $V(S) = \pi R^2$ . If  $\mathbf{X} = (X_1, X_2)$  is uniformly distributed over the circle, then by symmetry the marginal distributions of  $X_1$  and  $X_2$  will be the same. The power is then

$$P \approx P(S) = E \|\mathbf{X}\|^2 = E[X_1^2] + E[X_2^2] = 2E[X_1^2] \quad (14.16)$$

where

$$E[X_1^2] = \frac{1}{\pi R^2} \int_{-R}^R x_1^2 \int_{-\sqrt{R^2-x_1^2}}^{\sqrt{R^2-x_1^2}} dx_2 dx_1. \quad (14.17)$$

This integral readily evaluates to  $R^2/4$ , and thus  $P(S) = R^2/2$ . □

The continuous approximation will now be used to study the coding and shaping gain of lattice codes.

### Shaping and Coding Gain

The coding gain is a function of the relative spacing of points in  $\Lambda$ , but is independent of  $S$ . The transmit power, and hence shaping gain, is a function of both  $\Lambda$  and  $S$ , but the continuous approximation discards the dependence on  $\Lambda$ . Based on this simplification, we will now characterize the shaping and coding gains.

The continuous approximation gives a useful approximation for  $\gamma$ , which in turn summarizes the error probability of the constellation based on a lattice code. From

(14.10),

(14.12) and (14.14),

$$\gamma = \frac{(2^v - 1)d_{\min}^2(\Lambda)}{2P} \approx \frac{\left[ \left( \frac{V(S)}{V(\Lambda)} \right)^{2/N} - 1 \right] d_{\min}^2(\Lambda)}{4P(S)/N} = 3 \cdot \gamma_\Lambda \cdot \gamma_S, \quad (14.18)$$

where

$$\gamma_\Lambda = \frac{d_{\min}^2(\Lambda)}{V^{2/N}(\Lambda)}, \quad (14.19)$$

$$\gamma_S = \frac{NV^{2/N}(S)}{12 \cdot P(S)}. \quad (14.20)$$

The approximation for  $\gamma_S$  assumes that the unity term can be ignored, which will be accurate for large constellations. The motivation for the factor of 3 is that  $\gamma = 3$  for a two-dimensional square constellation, which we take as a baseline against which to compare other shapes. (It will be shown shortly that the multidimensional cube, which is a generalization of the two-dimensional rectangle, also has  $\gamma = 3$ .) We will see that both  $\gamma_S$  and  $\gamma_\Lambda$  are normalized to unity for a square QAM constellation, our reference constellation.

**Example 14-5.**

For the square constellation of Example 14-3, the shaping gain is

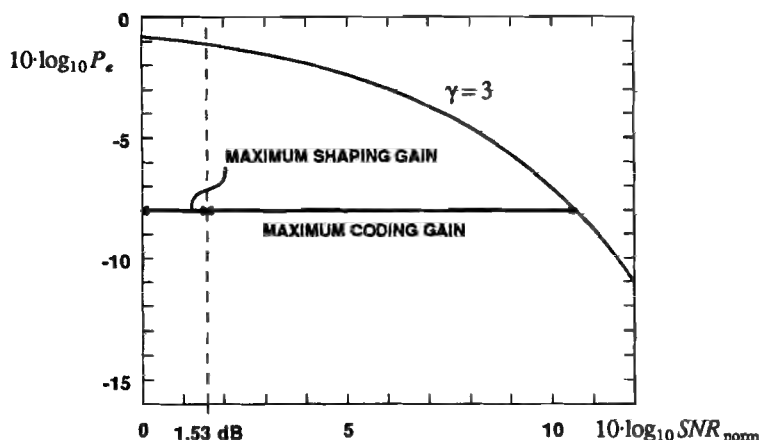
$$\gamma_\Lambda = \frac{2V(S)}{12P(S)} = \frac{2 \cdot 4R^2}{12 \cdot 2R^2/3} = 1. \quad (14.21)$$

Similarly, for a square lattice of points, if the spacing between adjacent points is  $d_{\min}$ , then the fundamental volume is  $V(\Lambda) = d_{\min}^2$  and thus  $\gamma_\Lambda = 1$ . Thus, for this case  $\gamma = 3$ , as we knew already.  $\square$

The factor  $\gamma_\Lambda$  includes all terms that are a function of  $\Lambda$ , and is defined as the *coding gain*. The factor  $\gamma_S$  includes all terms that are a function of  $S$ , and is defined as the *shaping gain*. For lattice codes with a large number of constellation points, the coding gain and the shaping gain are factors that can be manipulated independently of one another, the former by choosing  $\Lambda$  and the latter by choosing  $S$ .

The coding gain compares the minimum distance of a lattice  $\Lambda$  against a two-dimensional square lattice  $\Lambda_2$ , our reference lattice. Comparing two lattices is tricky since often both minimum distance and fundamental volume will be different, and the lattices may also have different dimensionality. A fair minimum distance comparison requires that the two lattices have the same density of points, that is, the same number of points per unit volume. If  $\Lambda$  and  $\Lambda_2$  are the same dimensionality and have the same shaping region  $S$ , then this implies the spectral efficiency will be the same (to accurate approximation).

An additional complication is that the two lattices may have different dimensionality. Thus, we scale the two lattices such that their density of points per two



**Figure 14-5.** The SNR gap to capacity for lattice codes with large constellations is divided into shaping gain and coding gain. The curve is  $P_e$  for a square QAM constellation ( $\gamma = 3$ ). The maximum shaping gain (for  $S$  an  $N$ -sphere as  $N \rightarrow \infty$ ) reduces the SNR gap to capacity by 1.53 dB.

dimensions is the same. This implies that, for the same shaping region, the codes based on the two lattices will have the same spectral efficiency and power. The power is the same because, in accordance with the continuous approximation, it is a function the shaping region only.

To compare  $\Lambda$  to the reference lattice  $\Lambda_2$ , first scale  $\Lambda_2$  by a factor  $\alpha$  to get a new square two-dimensional lattice  $\alpha \cdot \Lambda_2$  with minimum distance  $\alpha \cdot d_{\min}(\Lambda_2)$  and fundamental volume

$$V(\alpha \cdot \Lambda_2) = \alpha^2 d_{\min}^2(\Lambda_2). \quad (14.22)$$

For a square lattice, the fundamental volume is the square of the minimum distance. Second, determine the fundamental volume of  $\Lambda$  per two dimensions, and then set it equal to  $V(\alpha \cdot \Lambda_2)$  so that both lattices have the same density of points per two dimensions. According to the continuous approximation, the number of points in  $\Lambda$  per two dimensions is

$$2^v = \frac{V^{2/N}(S)}{V^{2/N}(\Lambda)}, \quad (14.23)$$

implying that the volume of  $S$  per two dimensions is  $V^{2/N}(S)$  and the fundamental volume of  $\Lambda$  per two dimensions is  $V^{2/N}(\Lambda)$ . Finally, set the fundamental volume per two dimensions of  $\alpha \cdot \Lambda_2$  and  $\Lambda$  equal,

$$\alpha^2 d_{\min}^2(\Lambda_2) = V^{2/N}(\Lambda), \quad (14.24)$$

which defines the scaling factor  $\alpha$ . Now that these fundamental volumes (and hence spectral efficiency for any  $S$ ) are equal, the coding gain is the ratio of the square of the minimum distance of  $\Lambda$  to that of  $\alpha \cdot \Lambda_2$ ,



$$\gamma_{\Lambda} = \frac{d_{\min}^2(\Lambda)}{\alpha^2 d_{\min}^2(\Lambda_2)} = \frac{d_{\min}^2(\Lambda)}{V^{2/N}(\Lambda)}, \quad (14.25)$$

consistent with (14.19). This establishes an interpretation of  $\gamma_{\Lambda}$  as the ratio of the square of the minimum distance of  $\Lambda$  to that of a scaled version of  $\Lambda_2$  (the reference lattice), where the scaling forces the spectral efficiency of the two lattices to be the same for any  $S$ .

A slightly different interpretation of  $\gamma_{\Lambda}$  is to assume that  $\Lambda$  and  $\Lambda_2$  have the same minimum distances, and compare their fundamental volumes per two dimensions. If we assume that  $\Lambda_2$  has minimum distance  $d_{\min}(\Lambda)$ , then the fundamental volume of  $\Lambda_2$  is  $d_{\min}^2(\Lambda)$  (since it is a square lattice). Since the fundamental volume per two dimensions of  $\Lambda$  is  $V^{2/N}(\Lambda)$ , it follows that  $\gamma_{\Lambda}$  is precisely the ratio of the fundamental volume per two dimensions of  $\Lambda_2$  to that of  $\Lambda$ . Since the fundamental volume is inversely proportional to the number of points in the constellation per two dimensions (for the same shaping region), the coding gain is the ratio of the number of points in the constellations for equal minimum distances.

A similar interpretation can be applied to the shaping gain. Again, the idea is to compare the power of the shaping region  $S$  against the power of a two-dimensional square region  $S_2$ , where both  $S$  and  $S_2$  have the same volume per two dimensions. Let the reference shaping region  $S_2$  be an  $2R \times 2R$  two-dimensional square, with each of the two dimensions in the range  $x \in [-R, R]$ . The "radius"  $R$  will be chosen to force the spectral efficiency to be the same as  $S$ , which will occur if the two shapes have the same volume per two dimensions. Since  $S$  has volume per two dimensions  $V^{2/N}(S)$ , and  $S_2$  has volume  $(2R)^2$ ,  $R$  is chosen to satisfy

$$(2R)^2 = V^{2/N}(S). \quad (14.26)$$

The shaping gain  $\gamma_S$  is then the ratio of the power per two dimensions of  $S_2$  to that of  $S$ . The power of  $S$  per two dimensions is  $2/N$  times the power  $P(S)$  per  $N$  dimensions. Furthermore, from Example 14-3,  $P(S_2) = 2R^2/3$ . Thus, the shaping gain is

$$\gamma_S = \frac{2R^2/3}{2P(S)/N} = \frac{N V^{2/N}(S)}{12P(S)}, \quad (14.27)$$

consistent with (14.20). This establishes an interpretation of  $\gamma_S$  as the ratio of the power per two dimensions of a two-dimensional square shape to the power per two dimensions for  $S$ , where the square shape is scaled so that the volumes per two dimensions (and hence spectral efficiencies) are the same.

It will be shown shortly that the maximum possible shaping gain as  $N \rightarrow \infty$  is  $10 \log_{10} \gamma_{S, \max} = 1.53$  dB. As a result, the SNR gap to capacity for lattice codes can be divided into two distinct parts, as shown in Figure 14-5. The  $\gamma = 3$  curve shows the SNR gap to capacity for a square two-dimensional constellation, where  $\gamma_{\Lambda} = \gamma_S = 1$ . The maximum shaping gain reduces the SNR gap to capacity, and the remaining SNR gap, labeled the "maximum coding gain", is potentially reduced by the choice of the code. A result of de Buda [1] shows that asymptotically as  $N \rightarrow \infty$  there exist lattice codes that achieve channel capacity, in the sense that they drive  $P_e$  to zero for any bit rate below the Shannon limit. (While there are flaws in the original proof of de Buda,

a new proof has been provided by Loeliger.) This answers a long-standing question in information theory; namely, is it possible to achieve the channel capacity limits with codes that have a structure that allows them to be implemented? The answer is yes, although unfortunately it is still not practical to implement soft decoding for lattice codes at high dimensionality.

### Cartesian Product Constellations

When we use a two-dimensional constellation and transmit a sequence of  $K$  symbols drawn from this constellation, we can consider that sequence as a vector in  $2K$ -dimensional Euclidean space. This sequence is actually drawn from a multidimensional constellation consisting of the Cartesian product of  $K$  two-dimensional constellations. Provided that coding and shaping gains are properly defined, we would expect that the coding and shaping gains of this  $2K$ -dimensional constellation would be equal to the coding and shaping gains of the underlying two-dimensional constellation. This is true, and in fact, we will now prove a more general result: given any lattice code  $C = (\Lambda + \mathbf{a}) \cap S$ , the coding and shaping gains of any  $K$ -fold Cartesian product constellation  $C^K$  is the same those of  $C$ .

Let  $C = (\Lambda + \mathbf{a}) \cap S$  be a lattice code in  $L$ -dimensional Euclidean space. We must define the  $K$ -fold Cartesian product  $C^K$ . A vector  $\mathbf{x}$  that is a point in  $C^K$  is a vector in  $KL$ -dimensional Euclidean space of the form

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K), \quad (14.28)$$

where  $\{\mathbf{x}_i, 1 \leq i \leq K\}$  are each points in  $C$ .

First consider the coding gain of  $C^K$ . The coding gain is independent of  $S$ , so we can work with  $\Lambda$  and  $\Lambda^K$ , where the latter is an  $LK$ -dimensional lattice consisting of a  $K$ -fold Cartesian product of  $L$ -dimensional lattices  $\Lambda$ . (It is simple to see that  $\Lambda^K$  is in fact a lattice.) The distance-squared between  $\mathbf{x} \in \Lambda^K$  and  $\mathbf{y} \in \Lambda^K$ ,  $\mathbf{x} \neq \mathbf{y}$ , can be written as

$$\|\mathbf{x} - \mathbf{y}\|^2 = \sum_{i=1}^K \|\mathbf{x}_i - \mathbf{y}_i\|^2, \quad (14.29)$$

where the  $\mathbf{x}_i, \mathbf{y}_i \in \Lambda$ . Since all the  $\mathbf{x}_i$  can be chosen independently of one another, and similarly the  $\mathbf{y}_i$ , the minimum distance will occur when all terms but one are zero, and the minimum of the one non-zero term is  $d_{\min}(\Lambda)$ . Thus,

$$d_{\min}(\Lambda^K) = d_{\min}(\Lambda), \quad (14.30)$$

so the Cartesian product does not affect the minimum distance. Likewise, we can show that the fundamental volumes per two dimensions are the same,

$$V^{2LK}(\Lambda^K) = V^{2L}(\Lambda), \quad (14.31)$$

establishing that the coding gains are identical for  $C$  and  $C^K$ . This follows directly from the following exercise, with  $U$  equal to the region corresponding to the fundamental volume of  $\Lambda$ .

**Exercise 14-1.**

Let  $U$  be a region of  $L$ -dimensional Euclidean space, and let  $U^K$  be a region that is the  $K$ -fold Cartesian product of  $U$ .

- (a) Show that  $V(U^K) = V^K(U)$ .
- (b) Let  $\mathbf{X}$  be a uniformly distributed random vector on region  $U^K$ . Show that its variance is  $P(U^K) = K \cdot P(U)$ .  $\square$

We can also show that the shaping gain of a Cartesian-product region  $S^K$  is

$$\gamma_{S^K} = \frac{LK V^{2LK}(S^K)}{12 P(S^K)} = \frac{LK V^{2L}(S)}{12 K P(S)} = \frac{L V^{2L}(S)}{12 P(S)} = \gamma_S. \quad (14.32)$$

In words, the shaping gain of  $C^K$  is equal to the shaping gain of  $C$ .

This result is fundamental to the understanding of multidimensional constellations (and more generally signal-space coding). Taking a Cartesian product of lattice codes does not affect the coding or the shaping gain. The way to achieve an increase in coding and shaping gains as the dimensionality is increased is to choose the components of the code dependently.

**Maximum Shaping Gain**

Figure 14-5 shows the SNR gap to capacity at any  $P_e$ . We now know that  $\gamma$ , and hence this gap, is divided between two factors, the coding gain and the shaping gain. The question arises as to how much gain is available from coding and how much is available from shaping. The answer is that the shaping gain is limited to 1.53 dB, which establishes that the remainder must be the available coding gain. We will now derive this result.

The maximum shaping gain is achieved by a spherical multidimensional constellation, as seen in the following argument. For a fixed  $\Lambda$ , a shaping region  $S$  that is not spherical and an  $N$ -sphere with the same volume will have, to accurate approximation, approximately the same number of lattice points and hence the same spectral efficiency. The continuous-approximation uniform density has the same height in both cases, since the volume is the same. If we move that region of  $S$  that is outside the sphere to the inside, the power will be reduced, since  $\|\mathbf{x}\|^2$  will be made smaller over that region.

The maximum shaping gain, that of the  $N$ -sphere, can be determined as follows. Define  $S_N(R)$  to be an  $N$ -dimensional sphere of radius  $R$ , and let  $\mathbf{x}$  be an  $N$ -dimensional vector with real-valued components. Then

$$S_N(R) = \{\mathbf{x} : \|\mathbf{x}\| \leq R\} = \left\{ \mathbf{x} : \sum_{i=1}^N x_i^2 \leq R^2 \right\}, \quad (14.33)$$

and the volume is

$$V[S_N(R)] = \int_{S_N(R)} d\mathbf{x}. \quad (14.34)$$

Changing the variable of integration to  $\mathbf{r} = \mathbf{x}/R$ , this volume can be expressed in terms

of the volume of a unit sphere,

$$V[S_N(R)] = V[S_N(1)] \cdot R^N, \quad (14.35)$$

where  $V[S_N(1)]$  will be determined shortly.

Suppose that  $\mathbf{X}$  is a random vector uniformly distributed over  $S_N(R)$ ;  $\mathbf{X}$  is called a *spherically uniform* random vector. The probability density function of  $\mathbf{X}$ ,  $f_{\mathbf{X}}(\mathbf{x})$ , is  $V^{-1}[S_N(R)]$  for  $\mathbf{x} \in S_N(R)$  and zero elsewhere. The marginal density of one component of  $\mathbf{X}$ , say  $X_1$ , can then be calculated. This marginal density  $f_{X_1}(x_1)$  is  $f_{\mathbf{X}}(\mathbf{x})$ , a constant, integrated over the region

$$\left\{ \sum_{i=2}^N x_i^2 \leq R^2 - x_1^2 \right\}. \quad (14.36)$$

This region is itself an  $(N-1)$ -dimensional sphere  $S_{N-1}(\sqrt{R^2 - x_1^2})$ , and the integral is the volume of that sphere. Thus, the marginal density is

$$f_{X_1}(x_1) = \frac{V[S_{N-1}(\sqrt{R^2 - x_1^2})]}{V[S_N(R)]} = \frac{V[S_{N-1}(1)]}{V[S_N(1)]} \cdot \frac{1}{R} \left[ 1 - \left[ \frac{x_1}{R} \right]^2 \right]^{(N-1)/2}. \quad (14.37)$$

The marginal density allows us to determine the volume of a unit sphere. Since it must integrate to unity,

$$\frac{V[S_N(1)]}{V[S_{N-1}(1)]} = \int_{-1}^1 (1 - \rho^2)^{(N-1)/2} d\rho, \quad (14.38)$$

where the change of variables  $\rho = x_1/R$  has been performed. The volume of a circle ( $N = 2$ ) is  $\pi R^2$ , so  $V[S_2(1)] = \pi$ . Integral (14.38) is known as a *Beta function*, and can be evaluated in closed form for integer values of  $N$ . For even  $N$  (the case of greatest interest), the result is,

$$V[S_2(1)] = \pi, \quad \frac{V[S_N(1)]}{V[S_{N-2}(1)]} = \frac{2\pi}{N}, \quad V[S_N(1)] = \frac{\pi^{N/2}}{(N/2)!}, \quad N \text{ even}. \quad (14.39)$$

The power  $P[S_N(R)]$  can also be determined from this marginal density, since the components of a spherically uniform vector are clearly identically distributed,

$$\begin{aligned} P[S_N(R)] &= E[\|\mathbf{X}\|^2] = E\left[\sum_{i=1}^N X_i^2\right] = N \cdot E[X_1^2] \\ &= \frac{NR^2 V[S_{N-1}(1)]}{V[S_N(1)]} \int_{-1}^1 \rho^2 (1 - \rho^2)^{(N-1)/2} d\rho. \end{aligned} \quad (14.40)$$

Again, the integral can be evaluated for integer  $N$ ,

$$P[S_N(R)] = R^2 \cdot \frac{N}{N+2}. \quad (14.41)$$

The variance of one component of  $\mathbf{X}$  is thus  $R^2/(N+2)$ .

Finally, the shaping gain of an  $N$ -sphere can be determined from the volume and power; it is

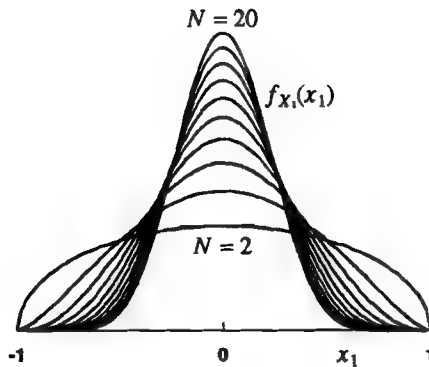
$$\gamma_{S_N(R)} = \frac{\pi(N+2)}{12[(N/2)!]^{2/N}}. \quad (14.42)$$

Using the Stirling approximation,  $k! \rightarrow (k/e)^k$  as  $k \rightarrow \infty$ , we get that  $\gamma_S \rightarrow \pi e/6$  asymptotically as  $N \rightarrow \infty$ . When this maximum shaping gain is achieved without coding gain, then the SNR gap to capacity is closed by  $10 \log_{10} \pi e/6 = 1.53$  dB, as shown in Figure 14-5. This is the best that shaping the multidimensional constellation can do.

### Marginal Density for Spherical Shaping

For two-dimensional constellations, we saw in Figure 14-3 that shaping resulted in a non-uniform marginal density. We can now explore the marginal density for higher dimensionality using the continuous approximation. The marginal density  $f_{X_1}(x_1)$  is plotted in Figure 14-6 for some even values of  $N$ , assuming spherical shaping. As the dimension  $N$  increases, the one-dimensional marginal density gets narrower, as also manifested by a decreasing variance  $R^2/(N+2)$ . (This is a natural consequence of spreading a total power of roughly  $R^2$  over an increasing number of dimensions.) Of more interest is the "bell shaped" appearance, similar to a Gaussian density, that emerges for large  $N$ .

Could it be that the density is in fact approaching Gaussian as  $N \rightarrow \infty$ ? To find out, define a normalized unit-variance random variable  $Y_1 = X_1 \sqrt{N+2}/R$ . Then  $Y_1$  has a density with the same shape, but its maximum value is  $\sqrt{N+2}$  rather than  $R$ . Its density is (for some appropriate constant  $C$ ),



**Figure 14-6.** The first-order marginal density of any component of a spherically uniform random vector is plotted for  $R = 1$  and even  $N$ .

$$f_{Y_1}(y_1) = C \cdot \left[ 1 - \left( \frac{y_1}{\sqrt{N+2}} \right)^2 \right]^{(N-1)/2}. \quad (14.43)$$

As  $N \rightarrow \infty$ , both  $(N+2)$  and  $(N-1)$  can be replaced by  $N$ , and

$$f_{Y_1}(y_1) \rightarrow C \cdot \left[ 1 - \frac{y_1^2}{N} \right]^{N/2} \rightarrow C \cdot e^{-y_1^2/2}, \quad (14.44)$$

where the limit  $(1 + x/k)^k \rightarrow e^x$  as  $k \rightarrow \infty$  has been used. Thus,  $X_1$  does approach Gaussian. (This is shown in [2] by a less direct conditional-entropy argument.) It can be shown by an identical method (Problem 14-7) that a  $K$ -dimensional marginal density of an  $N$ -dimensional spherically uniform random vector approaches a joint Gaussian density with independent components for fixed  $K$  as  $N \rightarrow \infty$ . In this sense, a spherically shaped lattice code approaches a white Gaussian source for large constellation sizes and high dimensionality.

Approaching channel capacity for the Gaussian channel requires that the transmitted signal be Gaussian. Multidimensional lattice codes with spherical shaping have approximately this property for high dimensionality in accordance with the continuous approximation. This is consistent with the de Buda result [1] that there exist lattice codes that approach capacity as  $N \rightarrow \infty$ .

### Relation of Spectral Efficiency and Power

The spectral efficiency of a constellation and its power are always directly related. For example, for a two-dimensional constellation, if we keep the minimum distance constant but increase the spectral efficiency by increasing the number of points, the constellation gets larger and the power increases. This basic tradeoff holds for multidimensional constellations as well, and it is important to quantify it.

First, consider the tradeoff between power and volume. In (14.18), since  $\gamma$  is not affected by any scaling of the signal constellation, and the coding gain  $\gamma_A$  is not a function of  $S$ , it follows that the shaping gain  $\gamma_S$  is independent of any scaling of  $S$ . Since

$$P(S) = \frac{N}{12 \cdot \gamma_S} V^{2/N}(S), \quad (14.45)$$

and the first factor is independent of any scaling of the region  $S$ , as  $S$  is scaled,  $P(S)$  is proportional to  $V^{2/N}(S)$ .

#### Example 14-6.

For an  $N$ -sphere with radius  $R$  and fixed dimension  $N$ , the volume is proportional to  $R^N$ , and the power is proportional to  $R^2$ , which is the volume raised to the power  $2/N$ .  $\square$

A fundamental relationship between spectral efficiency and signal power follows from (14.45). Substituting for the power per two dimensions,

$$P = 2P(S)/N \quad (14.46)$$

and using (14.12),

$$P = \frac{V^{2/N}(\Lambda)}{6\gamma_s} \cdot 2^v. \quad (14.47)$$

Thus, for a fixed lattice  $\Lambda$  (and hence fixed coding gain), the power per two dimensions is proportional to  $2^v$ , where  $v$  is the spectral efficiency in bits per two dimensions (bits per complex symbol). Thus, if we add one bit per complex symbol while holding the coding gain constant, the power per complex symbol is doubled.

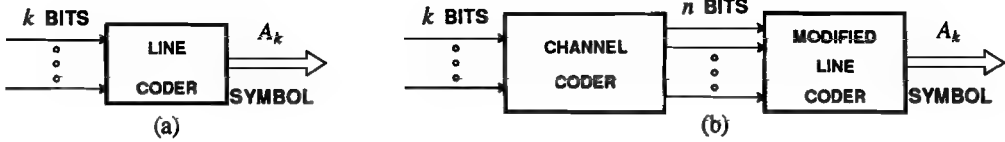
## 14.2. TRELLIS CODES

Lattice codes are useful for a modest number of dimensions. For large dimensionality, their implementation complexity becomes excessive, because the number of points in the signal constellation grows exponentially and the receiver multidimensional slicer becomes impractical to implement. Significant coding gains can be achieved with lower complexity by using an FSM in the transmitter (as was illustrated in Figure 14-1), possibly in conjunction with a multidimensional signal constellation. The lower complexity comes from the inherent simplicity of the transmitter FSM and the availability of the Viterbi algorithm for ML detection in the receiver.

The basic advantage of signal-space coding is the same for both approaches in Figure 14-1. Namely, by going to a higher dimensionality space we can increase the minimum distance in relation to the transmit signal power. In both cases, the sequence of data symbols is *not* a Cartesian product of two-dimensional symbols. Rather, the symbols are dependent on one another, and, as was seen in Section 14.1, this dependence is the essence of achieving coding and shaping gain. The FSM introduces dependence of the successive symbols by its symbol-to-symbol state memory. The coding gain due to the FSM can augment the coding and shaping gain due to constellation design. A signal-space coder based on an FSM is often called a *trellis coder*, because the FSM is conveniently represented by its trellis (Chapter 9).

The convolutional coder of Chapter 13 is a convenient FSM to use in a signal-space code. In Chapter 13 we thought of the additional bits introduced by the convolutional coder as increasing the bit rate. In signal-space coding, this redundancy is normally mapped into a larger symbol constellation, rather than an increased symbol rate. The bandwidth required for transmission is not increased, nor is total noise admitted by the receive filter. Of course, a penalty is paid in an increase in the number of points per multidimensional symbol, which taken by itself will either reduce the minimum distance or increase the transmitted power. However, the advantages of working in a multidimensional space more than makes up for this penalty.

A simple form of trellis coding proposed by Ungerboeck [3] uses a two-dimensional constellation, and is illustrated in Figure 14-7. In Figure 14-7a, an uncoded PAM data symbol is generated by a line coder (two-dimensional symbol constellation). The size of the constellation is  $2^k$ , and the information bit rate is  $k$  bits per symbol. The trellis coder of Figure 14-7b modifies this configuration by adding a rate  $k/n$  channel coder, for example based on the convolutional coder of Chapter 13. The line coder is modified to use a constellation of size  $2^n$ , where  $n > k$ . Significant



**Figure 14-7.** (a) An uncoded system transmitting  $k$  bits per two-dimensional symbol with a constellation of size  $2^k$ . (b) A signal-space trellis coder. The rate  $k/n$  channel coder introduces redundancy and the line coder accommodates that redundancy by using a constellation of size  $2^n$ .

coding gains can be achieved in this way.

#### Example 14-7.

A trellis code designed by Wei [4] is a crucial part of 9600 b/s voiceband data modems compatible with the CCITT V.32 standard. In that standard, the symbol rate is 2400 symbols per second, so a 16 point constellation would suffice without coding to support 9600 b/s. The standard uses a rate-2/3 convolutional code and a 32-point constellation. Most voiceband data modem standards faster than 4800 b/s depend on trellis codes.  $\square$

Assume that for a particular additive white Gaussian noise channel, an acceptable probability of error is achievable *without coding* at some SNR using a constellation of size  $M$ . Using coding we can reduce the SNR at the same error probability, or reduce the error probability at the same SNR; the improvement is limited by the Shannon channel capacity. The key observation made by Ungerboeck is that most of this theoretical reduction can be achieved using a constellation of size  $2M$  plus a channel coding algorithm. It is not greatly advantageous to use a constellation of size greater than  $2M$ , and it is not necessary to increase the symbol rate.

The justification for Ungerboeck's observation depends on Figure 4-4 and Figure 4-5. These figures show theoretical channel capacity (the maximum information rate for *error free* transmission) under various assumptions. In each figure, the left-most curve is the Shannon bound for discrete-time channels with additive Gaussian white noise. No assumption is made about the input constellation. The rest of the curves constrain the input constellation to be discrete-valued with equally likely symbols, and show the resulting channel capacity as a function of SNR. In Figure 4-4, for example, at low SNR we can get close to the Shannon bound using a four level PAM signal, 4-AM. As the SNR increases, with 4-AM the information rate cannot exceed two bits per symbol.

A set of dots are plotted on the curves that show the SNR at which a probability of error of  $10^{-5}$  is achievable for a particular constellation *without coding*. For example, from Figure 4-4 we see that at about 19 dB SNR we can achieve  $10^{-5}$  with a 4-AM constellation. Without coding, 4-AM transmits two bits per symbol. But *with coding*, using an 8-AM constellation we can theoretically transmit 2 bits per symbol (*error free*) down to about 13 dB SNR. Hence, using a coded 8-AM constellation, we should be able to design a code with a total gain (coding plus shaping gain) of  $19 - 13 = 6$  dB. Using larger constellations cannot improve the total gain by more



than about 1 dB, because the 6 dB gain is already so close to the Shannon bound. Furthermore, since there is no increase in bandwidth with coding, the gain is fully realized. There is no more noise for the coded system than for the uncoded system, unlike some situations considered in Chapter 13, in which the additional noise offset some of the gain.

### 14.2.1. Simple Trellis Codes

Designing trellis codes with total gains of 3 to 4.5 dB is easy, and the resulting codes are reasonably easy to implement. Simple trellis codes consist of convolutional coders followed by line coders that accommodate the redundancy with a larger alphabet. In this section we will make no attempt to separate the coding gain from the shaping gain. We will also make no attempt to separate the coding gain due to the FSM from the coding gain due to constellation design. We will instead evaluate the overall gain of some simple trellis codes by comparing them to uncoded systems that achieve the same overall bit rate in the same bandwidth. Then in Section 14.3 we develop a model that separates the gains from the various sources.

#### Example 14-8.

Consider the convolutional coder of Figure 13-8, reproduced with a line coder shown explicitly in Figure 14-8. The line coder is 4-PSK, as shown. The trellis is shown in Figure 14-9a. The only difference between this trellis and the one in Figure 13-11b is that the transitions are labeled  $(B_k, A_k)$  rather than  $(B_k, [C_k^{(1)}, C_k^{(2)}])$ . (Two slightly simpler but equivalent trellis codes are given in Problem 14-9 and Problem 14-10.) The performance advantages apply equally to all three coders.  $\square$

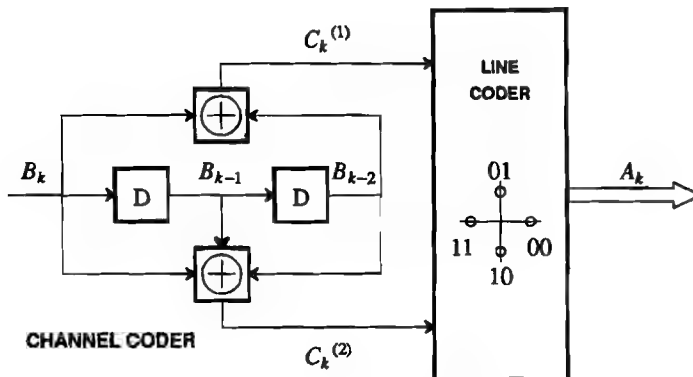


Figure 14-8. A trellis coder consisting of a convolutional coder followed by a line coder.

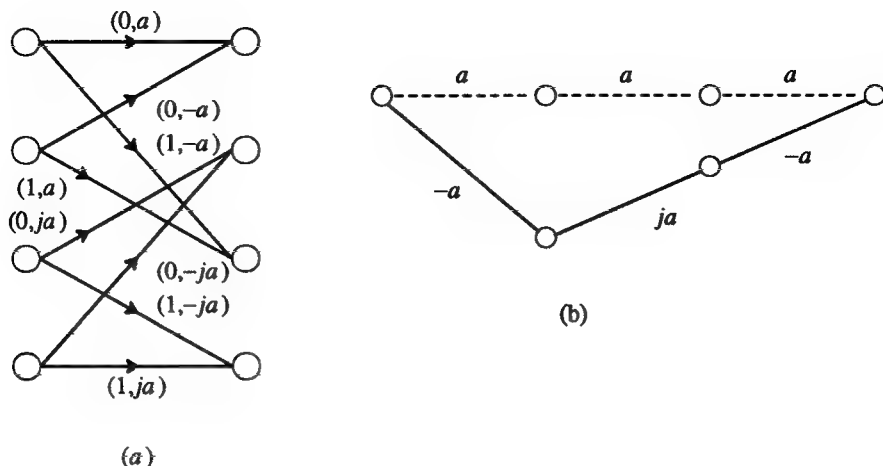


Figure 14-9. a. One stage of the trellis for the trellis code in Figure 14-8. b. The minimum-distance error event, assuming the correct state trajectory is the all-zero state trajectory.

### 14.2.2. Total Gain of Trellis Codes

The total gain that can be achieved with a trellis code depends on the number of states in the FSM. Roughly speaking, it is easy to get coding gains of about 3 dB with 4 states, 4.5 dB with 16 states, and close to 6 dB with 128 or more states [5]. In this section we illustrate how to determine the performance of a trellis code by directly comparing it to an uncoded system. In Section 14.3 we will show how shaping, constellation design, and the FSM individually contribute to this overall gain.

Because of the FSM convolutional coder, trellis codes lend themselves to soft decoding using the Viterbi algorithm. This type of decoding is the most common in practice, so we restrict our attention to it. In Appendix 9-C we showed that at high SNR

$$\Pr[\text{symbol error}] \approx KQ(d_{E,\min}/2\sigma) \quad (14.48)$$

where  $K$  is the error coefficient, lying between  $P$  and  $R$  defined by (9.158) and (9.150). Hence, as with soft decoded convolutional codes, the performance of trellis codes is dominated by the error events with the minimum Euclidean distance  $d_{E,\min}$ . However, caution is in order when calculating this distance. In general, trellis codes are not linear even when the underlying convolutional code is linear (see Appendix 13-A and Problem 14-11), so it is usually unsafe to simply find the error event closest to the zero state trajectory (see Problem 14-11). Instead, we may have to be more careful and systematically find the minimum distance error event for each possible correct path through the trellis, as done in Section 9.6. (The model of Section 14.3 separates the contribution of the FSM, and in most cases this leads to linearity and simplified methods for calculating minimum distance.)

**Example 14-9.**

In Figure 14-9b we show the minimum-distance error event assuming the correct state trajectory is zero; it has a distance of  $\sqrt{10}a$ . This distance is calculated using complex arithmetic,

$$|a + a|^2 + |a - ja|^2 + |a + a|^2 = 10a^2. \quad (14.49)$$

For this case it is easy to verify that there is no error event with distance less than  $\sqrt{10}a$ . Observe simply that all branches diverging from the same node have distance  $2a$  from each other, and all branches converging on the same node have distance  $2a$ , so the minimum distance is bounded from below by

$$d_{E,min} \geq \sqrt{(2a)^2 + (2a)^2} = \sqrt{8}a. \quad (14.50)$$

We can further determine that after two paths diverge, all possible combinations of subsequent branches have distance  $\sqrt{2}a$  so

$$d_{E,min} \geq \sqrt{(2a)^2 + (2a)^2 + 2a^2} = \sqrt{10}a. \quad (14.51)$$

Hence

$$d_{E,min} \geq \sqrt{10}a \quad (14.52)$$

for all possible state trajectories. It is also easy to verify that for all possible state trajectories there is exactly one error event at distance  $\sqrt{10}a$ .  $\square$

In Example 14-9 every state trajectory has exactly one error event at distance  $d_{E,min}$ , and this error event has exactly one symbol error, so  $K = P = R = 1$ . Consequently, at high SNR,

$$\Pr[\text{symbol error}] \approx Q\left[\frac{d_{E,min}}{2\sigma}\right]. \quad (14.53)$$

To compare this performance to an uncoded system, the noise variance  $\sigma^2$  is the same in both cases because the signal bandwidth is the same. We need to simply find an uncoded system with the same average transmit power that carries the same number of bits.

**Example 14-10.**

Continuing the previous example,  $d_{E,min} = \sqrt{10}a$  so

$$\Pr[\text{symbol error}] \approx Q\left[\frac{\sqrt{10}a}{2\sigma}\right]. \quad (14.54)$$

Using the pessimistic assumption that  $\Pr[\text{bit error}] \approx \Pr[\text{symbol error}]$  (see (9.165)), we get

$$\Pr[\text{bit error}] \approx Q\left[\frac{\sqrt{10}a}{2\sigma}\right]. \quad (14.55)$$

An uncoded 2-PSK system with alphabet  $\Omega_A = \pm a$  has the same transmit power as the coded 4-PSK system with alphabet  $\Omega_A = \{\pm a, \pm ja\}$  and carries the same number of source bits. It has a probability of error

$$\Pr[\text{bit error}] = Q(a/\sigma). \quad (14.56)$$

The coded system is better by approximately

$$20 \log (\sqrt{10}/2) \approx 4 \text{ dB} . \quad (14.57)$$

We have achieved the same improvement over the uncoded system as was achieved by convolutional coding with soft decoding, but without any increase in the bandwidth!  $\square$

In more complicated cases (where  $P \neq R$ ) we can find  $P$  and  $R$  to estimate  $K$  in (14.48), as illustrated in Appendix 9-C. It is more common, however, to assume that  $K$  is reasonably small and ignore it. A widely used rule of thumb is that at error rates on the order of  $10^{-5}$  or  $10^{-6}$ , if  $K$  is not too large, every increase in  $K$  by a factor of 2 costs about 0.2 dB of coding gain [6].

Trellis coding is usually preferred over convolutional or block coding for bandlimited channels with additive white Gaussian noise, unless severe nonlinearities or hardware complexity make the increased alphabet size impractical.

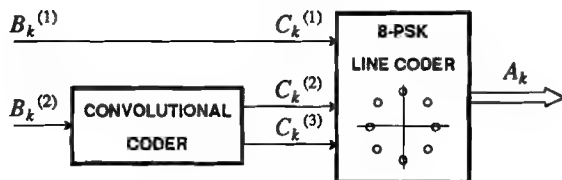
### 14.2.3. More Elaborate Trellis Codes

In the previous examples of trellis codes, one source bit was processed with a rate 1/2 convolutional coder to yield two coded bits. Representing the two coded bits requires an alphabet of size four. The technique is easily extended to make use of alphabets larger than four.

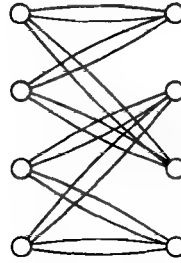
#### Example 14-11.

A simple extension of the trellis code described in the previous section is illustrated in Figure 14-10. This trellis code uses an alphabet of size eight, and can be built with any of the rate 1/2 convolutional coders in Figure 14-8, Problem 14-9, or Problem 14-10. Note that  $B_k^{(1)}$  has no effect on the shape of the trellis. The transition from one state to another is controlled entirely by  $B_k^{(2)}$ . Each transition from one state to another, therefore, occurs for two possible values of  $B_k^{(1)}$ , zero and one. This can be represented by showing two parallel transitions between every pair of states, as shown in Figure 14-11.  $\square$

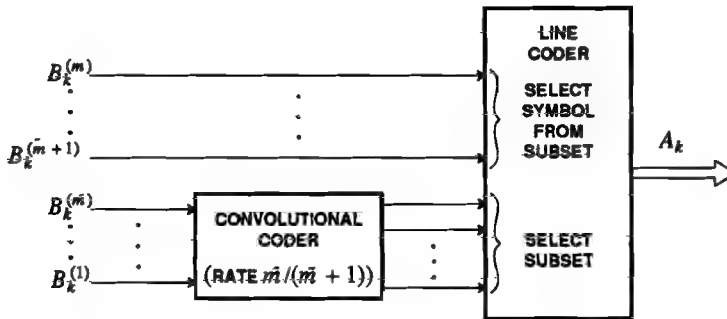
The general technique, illustrated in Figure 14-12, can be stated as follows. Given a channel with a bandwidth limitation, determine the symbol rate that can be transmitted. Determine the size  $2^m$  of the alphabet that would be required (without coding) to transmit the source bits at the desired bit rate. Then double the size of the alphabet to  $2^{m+1}$  and introduce a channel coder that produces one extra bit. The coder need not



**Figure 14-10.** A simple extension of the previous trellis code increases the number of source bits per symbol by using one uncoded bit and a larger alphabet.



**Figure 14-11.** The extra uncoded bit in Figure 14-10 can be represented in the trellis using parallel branches, as shown. One of two parallel branches is taken, depending on the value of the extra bit.



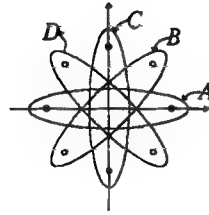
**Figure 14-12.** Trellis coders made by cascading a convolutional coder and a line coder can divide the incoming source bits into  $\tilde{m}$  bits to code and  $m - \tilde{m}$  bits to leave uncoded. The required convolutional coder has rate  $\tilde{m}/(\tilde{m} + 1)$ , and the trellis will have  $2^{(m - \tilde{m})}$  parallel transitions between every pair of states.

code all incoming bits, as shown in Figure 14-12. However, leaving some bits uncoded may affect performance.

A trellis with  $m - \tilde{m}$  uncoded bits has  $2^{m - \tilde{m}}$  parallel transitions between every pair of states. When there are parallel transitions, a very short error event consists of mistaking one of these parallel transitions for the correct one. The coding does not defend at all against this error event. To minimize its probability, we should ensure that the Euclidean distance between the symbols corresponding to parallel transitions is maximized.

#### Example 14-12.

To complete the design of the coder in Figure 14-10, we need to design the mapping of bits into 8-PSK symbols. Designing the mapping is the same as assigning symbols to transitions in the trellis Figure 14-11. Divide the 8-PSK constellation into four subsets, as shown in Figure 14-13. To ensure that parallel transitions in Figure 14-11 have symbols as far apart as possible, symbols for parallel transitions are selected from the same subset. Hence,



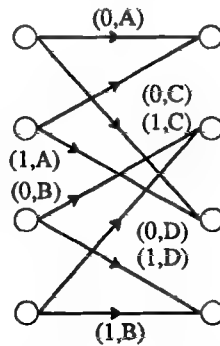
**Figure 14-13.** The 8-PSK symbol set is divided into four subsets. To ensure that parallel transitions in Figure 14-11 have symbols as far apart as possible, symbols for parallel transitions are selected from the same subset.

in Figure 14-10,  $C_k^{(2)}$  and  $C_k^{(3)}$  select the subset, A, B, C, or D, and  $C_k^{(1)} = B_k^{(1)}$  selects the point within the subset. Now assign subsets to pairs of transitions to try to maximize the minimum distance. A simple heuristic is to try and keep the distance between diverging or merging branches as large as possible. For example, C is the furthest subset from A, so the two pairs of branches emerging from state zero should be assigned subsets C and A. Using this rule, the mapping is shown in Figure 14-14.  $\square$

As usual, the performance of the code will be dominated by the error event with the smallest distance from the correct path. For coders with parallel transitions, we need to check the distance between parallel transitions to see if these are the closest error events.

#### Example 14-13.

Continuing the previous example, assume  $|A_k| = a$  for all symbols in the 8-PSK alphabet. Then using the mapping developed in the previous example, the distance between parallel transitions is  $2a$ . The next closest error event turns out to be at distance  $a\sqrt{6} - \sqrt{2} = 2.14a > 2a$ , so  $d_{E, \min} = 2a$  (see Problem 14-13). Hence the probability of an



**Figure 14-14.** Parallel branches (shown here as single lines) are assigned a subset of the signal set in such a way as to maximize the distance of diverging or merging branches.

error event is approximately

$$\Pr[\text{error event}] \approx Q\left[\frac{a}{\sigma}\right]. \quad (14.58)$$

These most probable error events result in exactly one bit error out of two (the uncoded bit), and there is only one such error event for each correct path, so we can assert

$$\Pr[\text{bit error}] \approx \frac{1}{2} Q\left[\frac{a}{\sigma}\right]. \quad (14.59)$$

To compare this to the uncoded system, note that the uncoded system requires an alphabet of size four to achieve the same source bit rate. Such an alphabet resulting in a signal with identical power is the 4-PSK alphabet  $\{\pm a, \pm ja\}$ . So, for the uncoded system,  $d_{E, \min} = \sqrt{2}a$ , and the probability of bit error is

$$\Pr[\text{bit error}] = Q\left[\frac{\sqrt{2}a}{2\sigma}\right]. \quad (14.60)$$

Ignoring the constant coefficient in (14.59), the total gain is

$$20 \log \sqrt{2} = 3 \text{ dB}. \quad (14.61)$$

Hence the parallel transitions degrade the performance by about 1 dB compared to the system in Figure 14-8, which does not have parallel transitions.  $\square$

## Mapping by Set Partitioning

In Example 14-13, we developed a mapping between the coded bits  $[C_k^{(1)}, C_k^{(2)}, C_k^{(3)}]$  and the transmitted symbols  $A_k$ . That mapping has the property that parallel transitions correspond to symbols that are as far apart as possible. A systematic way to design such mappings in general is known as *mapping by set partitioning*, proposed by Ungerboeck [3].

From bandwidth and bit-rate considerations we can determine the number of symbols required in the alphabet. If the number is  $2^m$  for an uncoded system, we have proposed using  $2^{m+1}$  for the coded system. However, we still have considerable freedom in choosing how to map the coded bits into symbols. The choice of mapping can drastically affect performance of the code. A good heuristic technique was proposed by Ungerboeck [3].

### Example 14-14.

In Figure 14-13 we divided an 8-PSK constellation into four subsets. Another way to view this partition is illustrated in Figure 14-15. The constellation is first divided into two subsets that maximize the distance within the subsets, and then subdivided again.  $\square$

The same principle can be applied to more elaborate constellations. A 16-QAM constellation is partitioned in Figure 14-16. Consider the trellis coder in Figure 14-12. It is indicated that the uncoded bits select a signal from the subset, and the coded bits select the subset. Hence there must be  $2^{m+1}$  subsets.

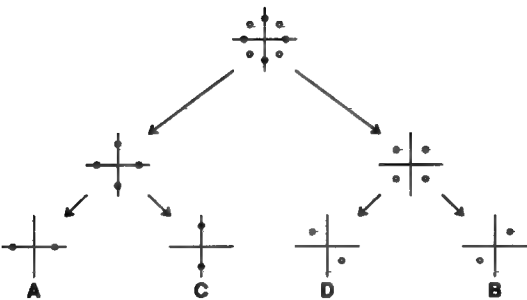


Figure 14-15. Systematic partitioning of an 8-PSK constellation.

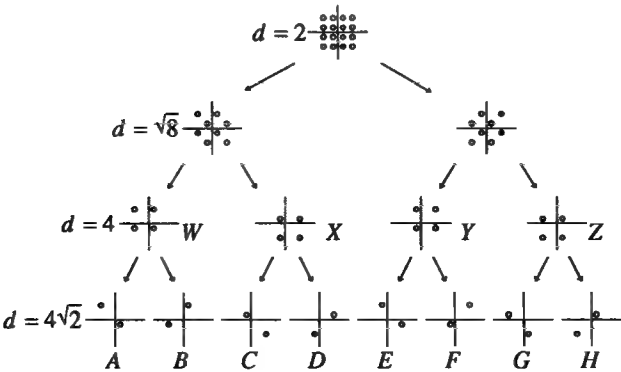


Figure 14-16. A 16-QAM constellation is partitioned into subsets so that the distance between the symbols within the subset is maximized.

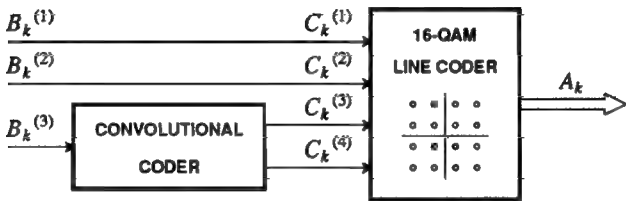


Figure 14-17. A trellis coder with  $\tilde{m} = 1$  and two uncoded bits.

**Example 14-15.**

If  $\tilde{m} = 1$  as in the previous examples, then 4 subsets are required. Thus the subsets in the third row of Figure 14-16 can be used in the coder shown in Figure 14-17. The uncoded bits select the symbol within the subset, and since there are two uncoded bits, each subset requires 4 symbols.  $\square$



To use the subsets in the fourth row in Figure 14-16, we need a trellis coder with  $\bar{m} = 2$ .

#### Example 14-16.

A trellis coder with one uncoded bit and two coded bits that uses the subsets in the fourth row in Figure 14-16 is shown in Figure 14-18. The convolutional coder is from Figure 13-18b. It is an 8-state systematic rate 2/3 convolutional coder of the feedback type. A reasonable mapping between the coded bits and the subsets from the fourth row of Figure 14-16 is shown in Figure 14-19. The total gain of this coder is approximately 5.33 dB [7]. We could of course add one more uncoded bit and use a 32-point cross constellation, in which case the total gain reduces to about 3.98 dB. Adding yet one more uncoded bit and using a 64-point QAM constellation reduces the total gain to about 3.77 dB (compared to a 32 point cross constellation used without coding).

It should be emphasized that these total gain coding gains compare the overall performance of the given trellis coder against an appropriate uncoded system. Not all the gain is due to the redundancy introduced by the convolutional coder; some of it is due to the constellations chosen for the comparison. This issue is addressed further in Section 14.3, where absolute measures of coding gain are developed.  $\square$

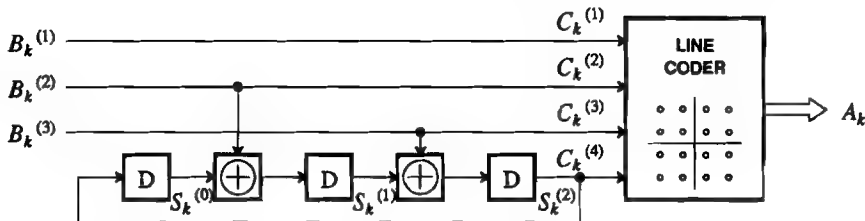
The general rules for mapping by set partitioning are:

- First, maximize the distance between parallel transitions.
- Next, maximize the distance between transitions originating or ending in the same state.
- Finally, use all symbols with equal frequency.

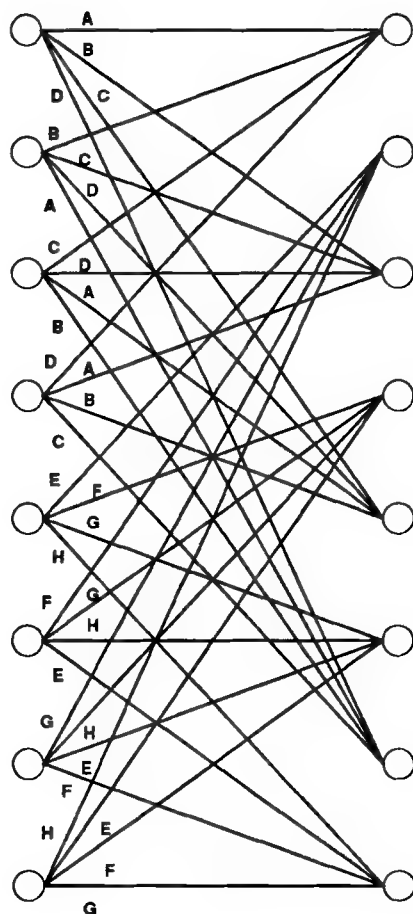
These rules are heuristic; they do not necessarily lead to a code that is optimal in any sense. In Section 14.3, we develop a more systematic approach to constellation partitioning based on cosets of a lattice.

### Catastrophic Codes

Considering only minimum-distance error events has its hazards. In most of our examples, we argued that there was only one minimum distance error event, and consequently it dominates the performance. In this section we give an example at the



**Figure 14-18.** A trellis coder with one uncoded bit and two coded. Also note that the convolutional coder is a feedback type, which is commonly used.

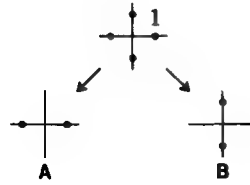


**Figure 14-19.** A mapping between the coded bits of Figure 14-18 and the subsets of the 16-QAM constellation in the fourth row of Figure 14-16.

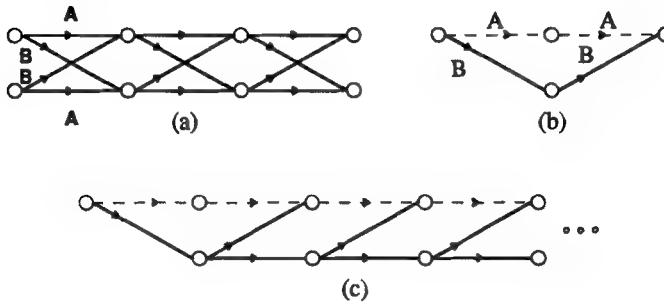
other extreme, where there are an infinite number of minimum-distance error events. More specifically,  $R$  in (9.150) may not be bounded, so  $K$  in (14.48) also may not be bounded. Such a code is called *catastrophic*.

#### Example 14-17.

A four-point constellation can be partitioned as shown in Figure 14-20. Consider transmitting 2 bits per symbol, using one bit to select subset A or B, and the other bit to select the point within the subset. This can be represented with a two-state trellis, as shown in Figure 14-21. A minimum-distance error event is illustrated in Figure 14-21b. The minimum distance is 2, which is 3 dB better than the minimum distance of  $\sqrt{2}$  in the uncoded system. If we assume that this is the only probable error event, then the total gain is about 3 dB. But we cannot expect this total gain. There are an infinite number of error events with the same minimum distance, some of which are shown in Figure 14-21c. In fact, it is easy to show



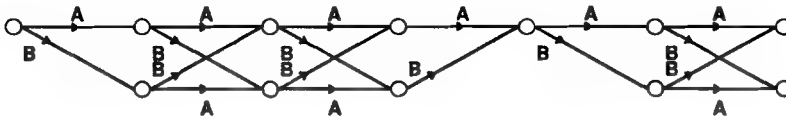
**Figure 14-20.** Set partitioning for a four point constellation.



**Figure 14-21.** a. A trellis for the code in Example 14-17. b. A minimum-distance error event. c. A set of minimum-distance error events. There are an infinite number of minimum-distance error events, so this code is catastrophic.

that  $R$  in (9.150) is unbounded (see Problem 14-16). We should not expect any gain at all, because we are representing two source bits with an alphabet of size four, so there is no redundancy!  $\square$

Catastrophic codes have the property that there are error events with finite distance but infinite length. The decoder may get into an error event during a burst of channel noise, and not get out again for a long time, especially if the channel quality improves! This will cause an infinite number of decoding errors. An interesting way of correcting the problem is illustrated in the following example.



**Figure 14-22.** The trellis of Figure 14-21a is modified so that it is forced to return to the zero state every fourth symbol. This converts the code to a block code, and reduces the message bit rate.

**Example 14-18.**

Suppose that the two-state trellis is forced to return to state zero every fourth symbol, as shown in Figure 14-22. Note that in the fourth symbol interval the coder does not have a choice of set A or B. Thus only one bit instead of two can be transmitted in the fourth symbol interval. In fact, the resulting code is a simple parity-check block code! Assuming the rate loss is not important, we can compare its performance with that of uncoded 4-PSK at high SNR. Now there are at most three error events with minimum distance starting at any given time, so the probability of a minimum-distance error event can be approximated as  $3Q(d_{E,min}/2\sigma)$  where  $d_{E,min} = 2$ . The factor of 3 is not important at high SNR, so nearly the full 3 dB improvement is realized. The code transmits only an average of 7/4 source bits per symbol, so there is some room for redundancy in the 4-PSK alphabet.  $\square$

Any trellis code can be converted into a block code of block size  $n$  by forcing the trellis to pass through a particular state every  $n$  symbols.

**Trellis Codes using Nonlinear Convolutional Codes**

In practice, there is little reason to restrict ourselves to linear convolutional coders (see Appendix 13-A). Some desirable trellis codes use nonlinear convolutional coders.

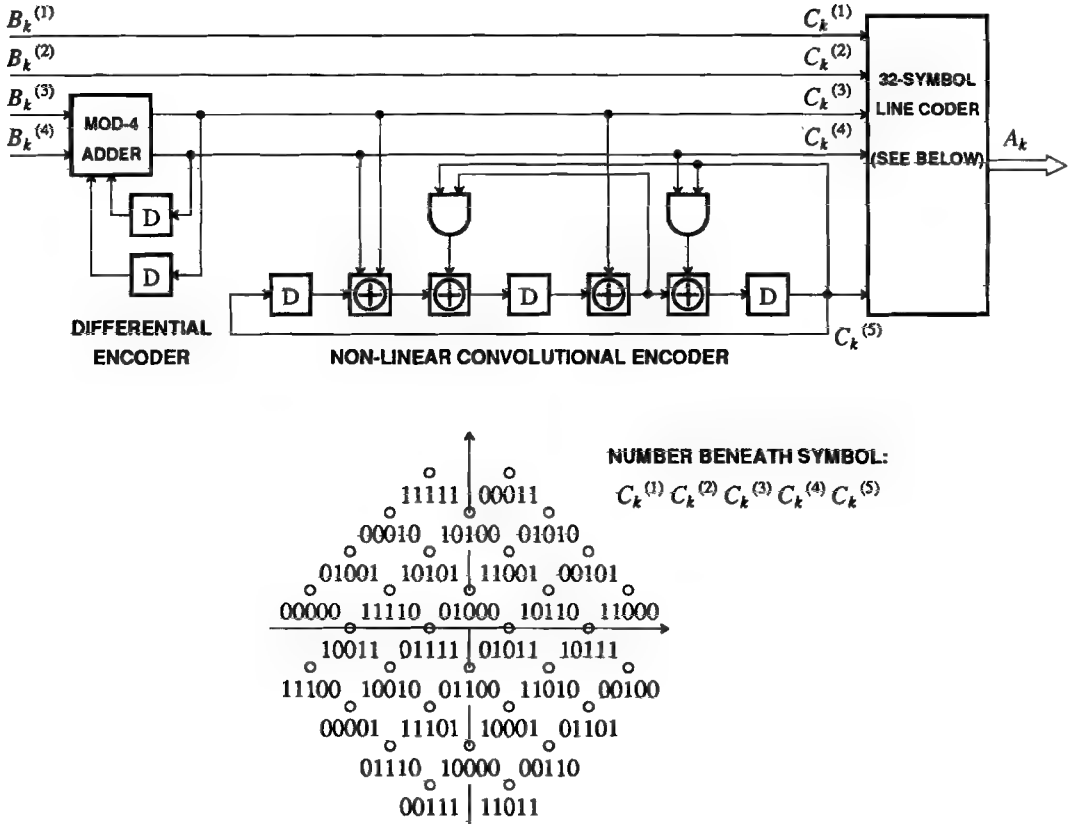
**Example 14-19.**

A trellis code using a nonlinear convolutional coder is shown in Figure 14-23. The coder was invented by Wei [4], and has been adopted in CCITT recommendation V.32 for voiceband modems operating at 9600 bits per second. Its main advantage is that the code is invariant under 90 degree phase shifts. This advantage will become clearer when we discuss carrier recovery in Chapter 16. Some of the symmetry is evident from examining the constellation in Figure 14-23. It is evident that  $C_k^{(1)}$  and  $C_k^{(2)}$  are the same at all points 90 degrees apart. Furthermore,  $C_k^{(3)}$  and  $C_k^{(4)}$  differentially encode the quadrant (see Chapter 16). That is, to move  $-90$  degrees, simply add one (modulo-four) to these two bits. This differential encoding is deliberately introduced by the differential encoder in the figure. The symmetry of the final bit is more subtle, but can be seen as symmetry in the trellis [4].  $\square$

**Multidimensional Trellis Codes**

In Section 14.1, the multidimensional signal constellation was described. It is realized by grouping one- or two-dimensional data symbols and treating them as a multidimensional vector. The general idea of trellis codes was presented in Figure 14-1 in terms of a multidimensional constellation, but all the examples thus far are two-dimensional.

In the context of a trellis code, the benefit of using a multidimensional constellation can be explained as follows. Recall the observation that doubling the symbol alphabet is sufficient to achieve almost all the available coding gain determined by the Shannon limit. However, doubling the size of the constellation, for the same minimum distance, will increase the signal power. The coding gain must overcome this immediate disadvantage. In Section 14.1, the continuous approximation predicted that doubling the size of the constellation increases  $P$ , the signal power per two dimensions, by  $2^{2/N}$  for an  $N$ -dimensional constellation. Thus, as the dimension of the constellation increases, the power penalty decreases. Expressed in dB, this penalty



**Figure 14-23.** A nonlinear convolutional coder uses modulo-two multiplication (and-gates) in addition to adders and delays. The example shown here is from the CCITT recommendation V.32.

is

$$10 \cdot \log_{10}(2^{2/N}) \approx 6/N. \quad (14.62)$$

Thus, it decreases from 3 dB for a two-dimensional constellation to just 1 dB for a six-dimensional constellation.

#### Exercise 14-2.

Show that the 32-point cross constellation in Figure 6-28 has 3 dB more power than the 16-point QAM constellation in Figure 6-27. Assume the points are all of the form  $A_k = [a_1, a_2]$  where  $a_i \in \{\pm 5, \pm 3, \pm 1\}$ . Thus, as predicted by the continuous approximation, the signal power is increased by 3 dB for a doubling of the number of points in the constellation.  $\square$

**Exercise 14-3.**

Consider a four-dimensional symbol

$$A_k = [a_1, a_2, a_3, a_4] \quad (14.63)$$

where  $a_i \in \{\pm 3, \pm 1\}$ . Show that the average squared power of this alphabet is 20, assuming all symbols are equally likely. There are  $4^4 = 256$  symbols in this alphabet. Now construct a four-dimensional alphabet with 512 symbols by adding symbols of the form:

$$[\pm 5, \pm 1, \pm 1, \pm 1] \quad (14.64)$$

and all its permutations (for a total of 64 possibilities) and symbols of the form

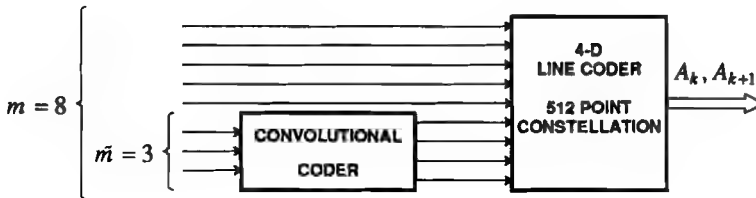
$$[\pm 5, \pm 3, \pm 1, \pm 1] \quad (14.65)$$

and all its permutations (for a total of 192 possibilities). Show that the average squared power of the 512 point constellation is 27, only 1.3 dB more than the average squared power of the 256 point constellation. The continuous approximation predicts a 1.5 dB penalty.  $\square$

While four-dimensional constellations like the one derived in Exercise 14-3 are difficult to draw, they are easy to use. Given a two-dimensional modulation system (a passband system), the first two coefficients  $a_1$  and  $a_2$  of the four-dimensional symbol are transmitted in one symbol interval as the real and imaginary parts of a complex symbol, and the last two coefficients  $a_3$  and  $a_4$  are transmitted in the next symbol interval. It is now easy to construct a trellis coder that makes use of this.

**Example 14-20.**

Continuing the previous example, the four-dimensional alphabet has 512 symbols, and hence can represent 9 coded bits. The trellis coder in Figure 14-24 will do the job. Three bits are coded to get four bits, and five bits are used uncoded. Calderbank and Sloane proposed this configuration in 1985 [8], and Forney, *et. al* proposed a similar configuration one year earlier [9]. Using an 8-state convolutional coder, they found a total gain of about 4.7 dB, ignoring the error coefficient  $K$ . They compared this performance to a two-dimensional trellis coder with the same source bit rate (4 bits per symbol) which has a total gain of about 4 dB. This suggests that use of a multidimensional trellis code yields an



**Figure 14-24.** A four-dimensional trellis coder. For every set of 8 bits that come in at the left, one four-dimensional symbol is produced by the line coder. These are actually transmitted, however, as two successive two-dimensional symbols.

additional total gain of about 0.7 dB in this example. However, the error coefficient  $K$  in this case is large enough to nullify much of this advantage.  $\square$

Ungerboeck tabulates several possible four and eight-dimensional trellis coders and their performance [7]. Many good multidimensional trellis codes are given by Wei [10]. Forney [6] and Ungerboeck [7] also tabulate good multidimensional trellis codes and their properties.

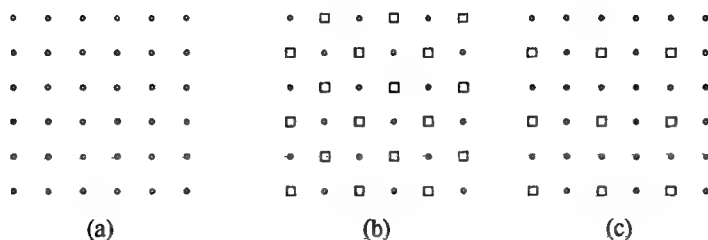
## 14.3. COSET CODES

In Section 14.2 we compared the performance of trellis coded systems against uncoded systems with the same transmit power and spectral efficiency. Although such comparisons are useful, they fail to properly account for the sources of improvement in performance. In particular, they mix coding gain with shaping gain, and they mix coding gain due to constellation design with coding gain due to the FSM. Moreover, the methods do not scale well to large constellations and to multidimensional constellations.

In this section, we introduce a more systematic approach based on the cosets of a lattice, as introduced by Calderbank and Sloane [11]. Trellis codes based on this coset partition are called *coset codes* by Forney [6]. Coset codes allow us to separate the coding gain due to the lattice, the shaping gain, and additional coding gain due to the FSM. Moreover, Ungerboeck's set partitioning generalizes to a simple systematic method that is easy to apply.

### 14.3.1. Lattice Partitions and Cosets

For a lattice  $\Lambda$ , a *sublattice*  $\Lambda'$  is a subset of the points in the lattice that is itself a lattice. Recall that a lattice is algebraically a group, meaning that it is closed under vector sums and differences. Any lattice must therefore include the zero point, and must be infinite in extent.



**Figure 14-25.** An illustration of portions of a lattice and two sublattices. Assume the lower left is the origin. (a) The integer lattice  $\mathbb{Z}^2$ . (b) The sublattice  $R\mathbb{Z}^2$  (defined by the squares). (c) The sublattice  $2\mathbb{Z}^2$  (also defined by the squares).

**Example 14-21.**

Let the one-dimensional integer lattice be written  $Z = \{\dots, -1, 0, 1, 2, \dots\}$ . Then the two-dimensional integer lattice in Figure 14-25a is the Cartesian product  $\Lambda = Z^2$ . A sublattice is defined by the squares in Figure 14-25b. Note that this sublattice is equal to a rotated and scaled version of the original lattice. Thus if  $\lambda$  is a vector representing a point in the original lattice (a two-dimensional vector with integer entries), then  $R\lambda$  is a vector representing a point in the sublattice, where

$$R = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (14.66)$$

Thus we write the sublattice  $\Lambda' = RZ^2$ . The sublattice defined by the squares in Figure 14-25c is simply a scaled version of  $\Lambda$ ,  $\Lambda'' = 2Z^2$ . Notice that  $2Z^2$  is a sublattice of  $RZ^2$ .  $\square$

Given a sublattice  $\Lambda'$  of  $\Lambda$ , a *coset* of  $\Lambda'$  is the set

$$\{\lambda' + \mathbf{c}; \text{ for all } \lambda' \in \Lambda'\} \quad (14.67)$$

for some  $\mathbf{c} \in \Lambda$ . Since  $\mathbf{c}$  identifies the coset, it is called the *coset representative*. Each coset is written  $\Lambda' + \mathbf{c}$ .

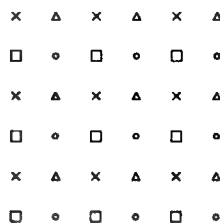
**Example 14-22.**

The sublattice  $\Lambda'' = 2Z^2$  in Figure 14-25c has a total of four unique cosets, shown in Figure 14-26. The coset representatives are  $\mathbf{c} \in \{(0, 0), (1, 0), (0, 1), (1, 1)\}$ . Only the coset identified by squares, the one with  $\mathbf{c} = (0, 0)$ , is itself a lattice, since it is the only coset that includes the zero vector. Similarly, the sublattice  $\Lambda' = RZ^2$  in Figure 14-25b has two cosets.  $\square$

A *partition* of  $\Lambda$  induced by  $\Lambda'$ , written  $\Lambda/\Lambda'$ , is the set of all cosets of  $\Lambda'$  in  $\Lambda$ , including  $\Lambda'$  itself. The *order* of a lattice partition, written  $|\Lambda/\Lambda'|$ , is the number of distinct cosets, and is finite.

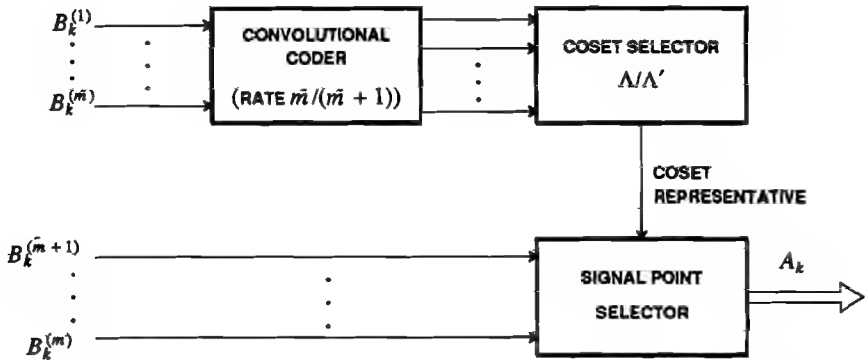
### 14.3.2. Application to Trellis Coding

In contrast to Figure 14-12, consider the view of trellis coding shown in Figure 14-27. The output of a convolutional coder (or more generally, an FSM) is used to select a coset from a partition  $\Lambda/\Lambda'$ . The FSM does not define which point to use within the coset. The output of the coset selector, therefore, is a coset representative.



**Figure 14-26.** A portion of four cosets of  $\Lambda'' = 2Z^2$  identified by four distinct shapes.





**Figure 14-27.** A view of trellis coding that enables deeper understanding of the sources of performance gain.

The point within the coset is selected by the "signal point selector", which therefore applies shaping. It might also translate the lattice by some vector  $\mathbf{a}$  so that the mean of the points it uses is zero. Thus, given a coset representative  $\mathbf{c}$ , the signal point selector chooses a constellation point in the set  $(\Lambda' + \mathbf{c} + \mathbf{a}) \cap S$ , where  $S$  is the shaping region and  $\mathbf{a}$  is a vector not constrained to lie on the lattice  $\Lambda$ .

The model in Figure 14-27 describes some trellis codes, but not all. The 8-PSK constellation of Figure 14-10, for example, cannot be described as cosets of a lattice. However, when constellations get large, practical considerations dictate using regular structures, and lattices have compelling advantages. Thus, the model in Figure 14-27 includes as special cases most practical examples of trellis codes with large constellations.

#### Example 14-23.

The trellis coders of Example 14-15, Example 14-16 and Example 14-7 can all be described in terms of Figure 14-27. Multidimensional trellis codes are also generally coset codes, except that the output of the signal point selector will be a vector of  $N/2$  complex symbols, rather than just a single complex symbol.  $\square$

There are potentially three distinct sources of performance gain in Figure 14-27:

- The lattice  $\Lambda$  may have coding gain, as discussed in Section 14.1.
- The signal point selector will use some shaping region  $S$  that may have shaping gain, also as discussed in Section 14.1.
- The convolutional coder will allow only particular sequences of cosets to be sent to the signal point selector, and thus introduces its own coding gain by increasing the minimum distance between sequences.

In Section 14.2, we made no attempt to separate the gain due to these three effects. The first two effects have been thoroughly studied in section 14.1, so the third is the only addition.

### 14.3.3. Coding Gain due to Redundancy

The convolutional coder in Figure 14-27 allows only a subset of all possible sequences of cosets. Thus, there is redundancy in such sequences. This means that the minimum distance between any two allowable sequences will be larger than the minimum distance between pairs of points in the lattice.

Let  $d_{\min}^2(C)$  be the minimum distance between any two sequences of cosets allowed by the convolutional coder  $C$ , where the distance between two cosets is taken to be the minimum distance between any point in one coset and any point in the other. This minimum distance will dominate the performance of the overall system. Let  $d_{\min}^2(\Lambda)$  be the minimum distance between any two points in  $\Lambda$ . Then the convolutional coder has increased the minimum distance by a factor of  $d_{\min}^2(C)/d_{\min}^2(\Lambda)$ .

There is a price paid, however, for this increase in minimum distance. Since there is typically one more bit emerging from the convolutional coder than going into it, twice as many points in the lattice will be needed within the shaping region to transmit the coded signal. Thus the size of the shaping region must increase. This *constellation expansion* reduces the gain, since it increases the power.

To quantify the effect of the constellation expansion, define the *redundancy*  $r(C)$  of the convolutional code  $C$  to be the number of redundant bits generated by the convolutional coder per  $N$  dimensions. In other words, it is the number of bits at the output of the convolutional coder minus the number of bits at its input. Usually  $r(C) = 1$ , as shown in Figure 14-27. Define the *normalized redundancy* (per two dimensions) as

$$\rho(C) = \frac{r(C)}{(N/2)}. \quad (14.68)$$

The transmitted power is increased by approximately  $2^{\rho(C)}$ . Note that with the typical  $r(C) = 1$ , the transmitted power is increased by  $2^{2/N}$ . Thus, as explained in Section 14.1, if  $N = 2$ , the power is doubled to accommodate the redundancy of the convolutional coder. This is intuitive, because the number of points in the transmitted (two-dimensional) constellation doubles. However, if  $N = 4$ , then the power increases by only a factor of  $\sqrt{2}$  because the number of points is doubled in a *four-dimensional* constellation, and this increase is divided between two successive two-dimensional symbols.

Combining the positive and negative effects of the convolutional coder, we get the coding gain due to the convolutional coder,

$$\gamma_C = \frac{d_{\min}^2(C)}{d_{\min}^2(\Lambda) 2^{\rho(C)}}. \quad (14.69)$$

This is simply the increase in minimum distance due to the convolutional coder divided by the increase in power (per two dimensions) due to the constellation expansion. We would expect the overall coding gain compared to an uncoded rectangular lattice with a rectangular shaping region to be, from (14.18),

$$\gamma = 3 \cdot \gamma_{\Lambda} \cdot \gamma_S \cdot \gamma_C, \quad (14.70)$$

where  $\gamma_\Lambda$  is the coding gain of the lattice defined in (14.19), and  $\gamma_S$  is the shaping gain of the lattice defined in (14.20). Thus, the convolutional coder adds additional coding gain  $\gamma_C$  on top of the lattice coding gain and the shaping gain. This is verified in the following exercise.

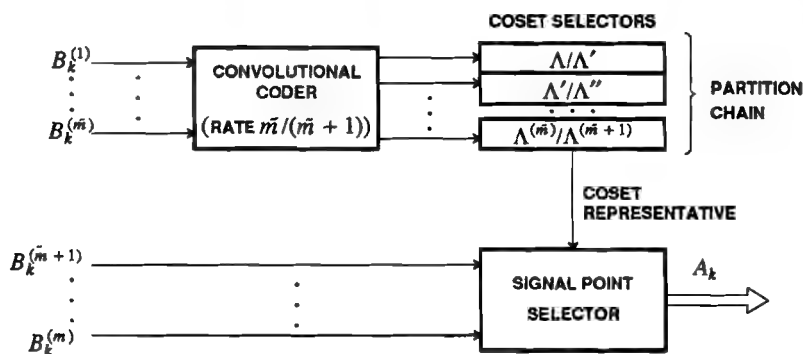
#### Exercise 14-4.

Show that (14.70) reduces to (14.10) where  $d_{\min}^2 = d_{\min}^2(C)$ . Hint: It might be helpful to use (14.46) and (14.12).  $\square$

The formulation in Figure 14-27 leads to another view of Ungerboeck's set partitioning, shown in Figure 14-28. Given a lattice  $\Lambda$ , form a sublattice  $\Lambda'$  of order two. For example, the sublattice  $RZ^2$  in Figure 14-25b is a sublattice of  $Z^2$  of order two. A coset induced by such a sublattice is therefore selected by one bit. Then form a sublattice  $\Lambda''$  of  $\Lambda'$  with order two. A coset for this partition is selected by another bit. Continue partitioning the lattice until there are enough partitions to encode all the bits from the convolutional encoder. For example, the set partitioning in Figure 14-16 is obtained by applying this procedure to  $\Lambda = Z^2$ .

## 14.4. SIGNAL-SPACE CODING AND ISI

Trellis coding is advantageous on many media where bandwidth is at a premium. On some, such as radio transmission between fixed antennas, or satellite transmission, ISI is not a significant problem. On others, like wire-pair and coaxial cable, ISI must be equalized or cancelled. The best techniques for countering ISI described in Chapters 10 and 11 are not immediately compatible with trellis coding or other signal-space codes. The reason is simply delayed decisions. In theory, the ML



**Figure 14-28.** Set partitioning can be viewed as a sequence of second order lattice partitions.

detector using the minimum-distance criterion may have to wait forever before it can make a decision. In practice, the Viterbi algorithm is used with some truncation depth (see Section 9.6). However, even a modest truncation depth introduces enough delay in the decision to compromise any decision-directed or decision-feedback technique. Thus, the straightforward combination of equalization for ISI with signal-space coding is not always possible. We now discuss some of these problems and ways around them.

### 14.4.1. Trellis Coding and Linear Equalization

Linear equalization (Chapter 10) is the simplest way to counter ISI. The LE is fortunately easy to combine with Viterbi decoding of a trellis code, since the input to the Viterbi detector is nominally free of ISI. The price paid for this simple solution, as opposed to ML detection or decision-feedback equalization, is higher noise enhancement.

There is potentially a problem when the adaptation of an LE is decision-directed (Chapter 11). A similar problem occurs with other decision-directed algorithms, such as carrier recovery (Chapter 16). Because of the delay in making decisions, the dynamics of the adaptation algorithm is altered. For example, in the receiver shown in Figure 6-23, the slicer is replaced with a sequence detector and the decisions are delayed. To ensure stability of the adaptation algorithms, their step sizes must be reduced, so the convergence time and tracking ability suffers. An alternative approach is to make tentative decisions with a conventional slicer, and use those tentative decisions to update the filter taps and carrier phase. Since these tentative decisions do not benefit from the coding gain, a minor degradation is suffered in the performance of the adaptive filters.

### 14.4.2. Trellis Coding and Transmitter Precoding

In Chapter 10, it was shown that decision-feedback equalization (DFE) results in less noise enhancement than the LE, since postcursor ISI is cancelled rather than equalized. The DFE depends on past decisions to cancel the postcursor ISI, and any delay in the availability of decisions due to sequence detection implies that only postcursor ISI with a compatible delay can be cancelled. For all practical purposes, this renders the DFE useless in the presence of ML trellis decoding, since it is typically the low-delay postcursor ISI that is most consequential. A way to combine the DFE and trellis coding, called "parallel decision-feedback equalization", will be discussed in the next subsection. Here we describe an alternative that is compatible with trellis coding.

Transmitter precoding (Section 10.1.4) was shown to achieve a performance essentially identical to the DFE by doing the postcursor cancellation in the transmitter rather than the receiver. The price paid is the need for knowledge of the channel response in the transmitter, a requirement that is compatible only with channels that are stationary or slowly time varying. This rules out, for example, rapidly fading radio channels.

Transmitter precoding was explained in the context of a one-dimensional signal constellation, although it is compatible with multidimensional constellations as well.

For simplicity, we will explain the combination of precoding with trellis coding for one-dimensional data symbols. Recall that if the data symbol  $a_k$  is chosen from an alphabet of size  $M$  (where  $M$  is even) consisting of odd integers less than  $M$  in magnitude, then the transmitter is designed in such a way that the *channel output* symbol is an extended data symbol

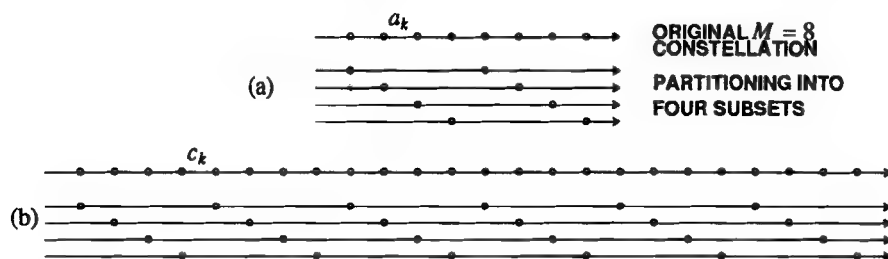
$$c_k = a_k + 2M \cdot i_k \quad (14.71)$$

where  $i_k$  is a sequence of integers chosen to minimize the peak transmitter power. The alphabet of  $c_k$  consists of all odd integers. This channel output data symbol is also corrupted by additive noise at the receiver, and in the absence of trellis coding is detected by applying an extended slicer that finds the closest odd integer to the received signal. The original data symbol  $a_k$  can be obtained uniquely from  $c_k$  by reducing it modulo  $2M$ .

Transmitter precoding is immediately compatible with trellis codes, if the constellation design and set partitioning is done in a compatible fashion. It requires no modification to the trellis coder, but the trellis decoder does have to be modified slightly. This is illustrated by an example.

#### Example 14-24.

An eight-point baseband constellation is shown in Figure 14-29a. It consists of all odd integers between -7 and 7, and can be partitioned using set partitioning into the four subsets shown. Given a rate one-half convolutional coder with one input bit, and a second input uncoded bit, the two convolutional coder output bits can be used to select the subset (from among four possibilities) and the uncoded bit can be used to select the point (from among two possibilities) within the selected subset. Suppose we are working with the extended constellation for  $c_k$  rather than  $a_k$ , as shown in Figure 14-29b. This extended subset is the lattice on which the constellation is based. Although  $c_k$  consists of all odd integers, it can still be partitioned into four subsets, the cosets induced by the specified partition. As before, the two coded bits select the subset, except that now the subsets are extended. The point within the subset, however, is chosen differently. The uncoded bit chooses one of the



**Figure 14-29.** Constellation set partitioning for an extended constellation. (a) Original  $M = 8$  constellation for  $a_k$ , consisting of all odd integers less than eight in magnitude. (b) An extended constellation for  $c_k$  consisting of all odd integers (only 24 points shown). Also shown in both cases is the partitioning into four subsets (corresponding to two coded bits).

two points closest to the origin, and then, after  $i_k$  is determined,  $2M \cdot i_k = 16 \cdot i_k$  is added to that choice. Adding  $16 \cdot i_k$  chooses one of the points in the extended partition.  $\square$

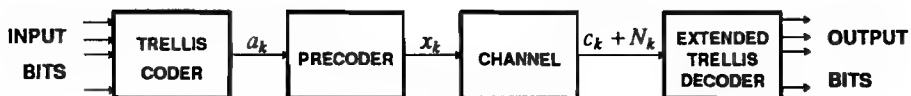
This example illustrates that, to be compatible with trellis coding, the symbol in an extended constellation is chosen in a two-stage process, as illustrated in Figure 14-30. The trellis coder is unchanged (except to make sure it is compatible with precoding). The output data symbol  $a_k$  with an  $M$ -ary alphabet is input to the transmitter precoder. The precoder output symbol  $x_k$  is determined so as to force the channel output signal component  $c_k = a_k + 2M \cdot i_k$  to be a point on the extended constellation, and at the same time minimize the peak power of the symbol  $x_k$ .

The trellis decoder must deal with an extended signal constellation. The trellis is unchanged, as determined by the trellis coder, and each pair of successive states in the trellis corresponds to a subset. The only difference is that each subset has an infinite set of points, extended by adding some unknown multiple of  $2M$ . There are now an infinite number of parallel branches between pairs of states, corresponding to the possible points in that subset. In principle, the ML detector must consider all parallel branches. In practice,  $i_k$  will be bounded, based on worst-case channel assumptions, so that only a finite number of parallel branches need be considered.

This particular example is easily extended to larger constellations, or to two or more dimensions. The trick is to start with a full lattice, rather than a lattice as limited by the shaping region, before performing the set partitioning. The mathematical framework of the sublattice and cosets can be helpful here. The number of sets in the partition is determined by the number of coded bits, and is typically four or perhaps eight. Finally, the transmitter trellis is designed assuming a cubic shaping, but the receiver decoder is designed assuming the full (extended) lattice.

### Coding Gains on Channels with ISI

It was established in Section 10.5.7 that if ISI is canceled by the DFE-ZF, then the SNR gap to capacity is the same at high SNR regardless of the nature of the ISI (including no ISI). Of course, the DFE-ZF is not directly compatible with trellis coding. However, in Section 10.1.4 it was shown that transmitter precoding can obtain essentially the same SNR at the slicer (or more generally decoder) input as the DFE-ZF. The conclusion is perhaps a surprising one; namely, the SNR gap to capacity at low probabilities of error can be closed to the same extent on channels with ISI as it can on channels that are free of ISI.



**Figure 14-30.** A combination of trellis coding with transmitter precoding for compensation of channel ISI.

This is not to imply that ISI has no impact on capacity; in fact, as illustrated in Section 10.5.6, ISI does have an impact on the capacity of the channel. Thus, the capacity is affected by the ISI, but what does not change is the SNR gap between uncoded square QAM (at a given  $P_e$ ) and capacity, and the extent to which that gap can be closed by coding and shaping gain.

Again it should be emphasized that this statement applies only at high SNR. The SNR gap to capacity is constant at all SNR for the DFE-MSE-U, even at low SNR. It is, however, difficult to exploit this property at low SNR because the combination of noise and residual ISI at the slicer is not white.

### Trellis Coding and Shaping Gain

Thus far, we have emphasized the application of trellis coding to obtaining coding gain. Trellis coding in conjunction with transmitter precoding can achieve shaping gain as well.

Recall from Section 10.1.4 that when we apply the continuous approximation to transmitter precoding, the conclusion is that the precoded symbols  $x_k$  are uniformly distributed and independent from symbol to symbol. That is, the shaping is cubic; there is no shaping gain. Thus, choosing  $i_k$  in the precoder to force the precoded sample  $x_k$  to obey  $\|x_k\| \leq M$  minimizes the energy of each precoded symbol, but doesn't minimize the average power of a sequence of symbols.

In fact, if the shaping is spherical, and the power per complex-symbol is kept constant at  $P$ , then the radius of an  $N$ -sphere is  $R = \sqrt{P \cdot (N/2 + 1)}$ . Since each coordinate is limited to  $R$ , as we move to spherical shaping the peak value of each coordinate actually increases (in proportion to  $\sqrt{N}$ ), and the marginal distribution approaches a truncated Gaussian. The conclusion is that to obtain shaping gains, we want to increase the allowed peak value of each symbol (which is precisely what we tried to avoid in the design of the transmitter precoder), and we want the distribution of the precoded symbols to be "Gaussian-like".

In light of these observations it is not surprising that shaping gain can be achieved by choosing the precoding  $i_k$  appropriately, and in particular *not* choosing them to minimize the peak value of the precoded symbol as we did in Chapter 10. A specific technique is proposed by Eyuboglu and Forney [12]. The basic idea is to choose the  $i_k$  to force the transmitted power averaged over time to be approximately equal to the allowed value, rather than the unshaped approach of minimizing the peak transmitted power of each individual symbol.

### 14.4.3. RSSD of Trellis Codes

On time-varying channels, the transmitter precoding requirement for knowledge of the channel response is problematic. In this case, there is an alternative, *reduced-state sequence detection (RSSD)* [13,14,15] in the receiver. With a trellis coder but no precoding in the transmitter, and channel ISI modeled as an FIR filter, the concatenation of the coder and the channel is an FSM signal-generation model. The received signal is the output of that FSM signal-generation model corrupted by noise. The ML detector for this signal-generation model is the Viterbi algorithm, which can in

principle be implemented assuming the channel impulse response is known to the receiver (but not necessarily the transmitter). The problem is the explosion in the number of states, which is the number of states in the trellis coder multiplied by the number of states in the channel model. RSSD reduces the complexity by retaining only a subset of the most important states.

The most attractive of these techniques essentially implements the Viterbi algorithm for the original trellis coder, but performs decision-feedback equalization on each path survivor in the trellis based on the history of that path. This is called *parallel decision-feedback equalization*. If there are  $n$  states in the trellis, then  $n$  distinct postcursor equalizer filters are used. Each filter uses the decisions from one of the  $n$  survivor paths to construct the next decoder input.

#### 14.4.4. Signal-Space Coding and Multicarrier Modulation

In the presence of ISI, multicarrier modulation (MCM) is an interesting alternative. In combined PAM and multipulse, the transmitted signal is represented by

$$S(t) = \sum_{k=-\infty}^{\infty} \sum_{n=0}^{N-1} A_{k,n} g_n(t - kT), \quad (14.72)$$

where the  $g_n(t - kT)$  are a set of  $N$  orthogonal waveforms (orthogonal for all values of  $n$  and  $k$ ). MCM is a special case where the pulses are time-limited sinusoids at equally spaced frequencies. As described in Section 6.9.1, in typical applications of MCM the bandwidth of the channel is kept constant, but the dimensionality of the signal set  $N$  is increased by increasing the symbol interval and decreasing the spacing between adjacent carriers.

Regardless of the particular choice of orthogonal pulses, if the MCM signal is transmitted through an additive white Gaussian noise channel, and matched filtering is applied at the receiver, then the equivalent channel model for the  $k$ -th symbol interval is a received vector of real- or complex-valued symbols  $(A_{k,0}, A_{k,1}, \dots, A_{k,N-1})$  corrupted by additive independent Gaussian random noise samples. This channel is mathematically indistinguishable from an ordinary PAM channel. However, with MCM there are more interesting degrees of freedom. For example, we can take the stream of data symbols corresponding to each dimension,  $\{A_{k,n}, -\infty < k < \infty\}$ , as an independent stream of data symbols for each  $0 \leq n \leq N-1$ , and independently apply a trellis coder/decoder to each one. Alternatively, we can serialize all the data symbols, generating  $\{A_{k,n}, 0 \leq n \leq N-1, -\infty < k < \infty\}$ , with a single trellis coder. In that case, the trellis coder is operating first across frequencies, and then across time.

As explained in Section 6.9, one of the benefits of MCM is that it offers inherent immunity to ISI, at least as  $N$  gets sufficiently large. The effect of ISI is twofold. First, it introduces dispersion within the data stream corresponding to each carrier. Second, it causes crosstalk between adjacent carriers, since they are typically overlapping in frequency and their orthogonality depends on a particular amplitude and phase characteristic. As  $N$  increases, each carrier is modulated at a lower symbol rate, and eventually the dispersion of each carrier becomes insignificant. Similarly, adjacent-carrier crosstalk will become insignificant as the distance between carrier frequencies shrinks, and the channel transfer function becomes essentially constant across the



bandwidth occupied by each pair of adjacent carriers.

Since MCM offers immunity to ISI, when  $N$  is chosen large enough it is also compatible with trellis coding (or other signal space coding). This is an alternative way to achieve significant coding and shaping gains on channels with ISI, at the expense of the delay associated with a long symbol interval.

## 14.5. FURTHER READING

The paper that established the importance of trellis coding is by Ungerboeck [3]. It followed a patent by Csajka and Ungerboeck [16]. Useful overviews including tables of trellis codes are [6,5,7,9]. An extensive treatment of trellis codes is given in the book by Biglieri, Divsalar, McLane, and Simon [17]. An alternative method of describing and specifying trellis codes is due to Calderbank and Mazo [18,19]. For an introduction to lattice codes, coset codes, and the combination of coding with ISI, the Dec. 1991 issue of the *IEEE Communications Magazine* is recommended, and the article by Forney and Eyuboglu is particularly helpful [20]. The August 1989 issue of *IEEE Journal on Selected Areas in Communications* includes a number of articles referenced in this chapter.

Some new and advanced techniques have been incorporated into the recent V.Fast modem standard [21], including an improved technique for shaping called *shell mapping* [22], and a new method for combining transmitter precoding with trellis coding and shaping known as *flexible precoding* [21].

This chapter has not covered the specialized topic of coding for magnetic recording channels, but [23] is an excellent introduction to recent results.

## PROBLEMS

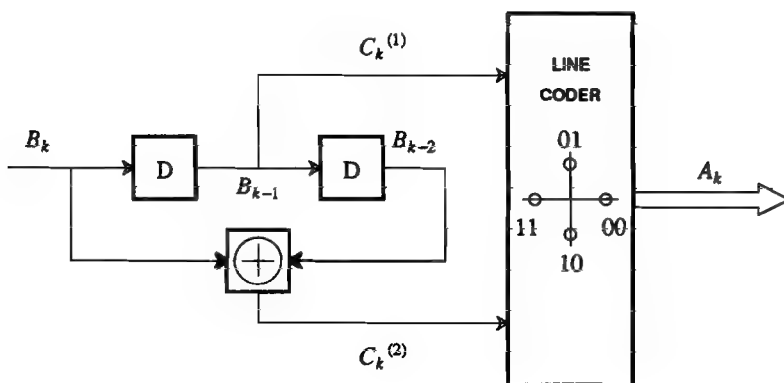
- 14-1. Describe how you would achieve two-dimensional shaping or coding gain (as in the circular or hexagonal constellations of Figure 14-2) in a baseband PAM system. Be sure to describe both the transmitter and receiver.
- 14-2.
  - (a) Calculate the continuous approximation for the shaping and coding gains for two-dimensional square lattice constellation with a circular shaping.
  - (b) Compare the results of (a) against a two-dimensional constellation with square shaping. What is the shaping gain in dB?
- 14-3. Calculate the coding gain  $\gamma_A$  for the two-dimensional hexagonal constellation of Figure 14-2. You can use the fact that the area of a hexagon with inscribed circle with radius  $r$  is  $6r^2 \tan(\pi/6)$ .
- 14-4. Show that the coding gain  $\gamma_A$  is invariant to the scaling the lattice  $\Lambda$ . Specifically, suppose  $\Lambda$  is scaled by multiplying all the basis vectors by a constant  $\alpha$ , and call the new scaled lattice  $\alpha \cdot \Lambda$ . Show that  $\gamma_{\alpha \cdot \Lambda} = \gamma_A$ .

- 14-5. Define  $C_N(R)$  as an  $N$ -cube that is  $2R$  on a side; that is, each dimension has range  $[-R, R]$ . This  $N$ -cube is the Cartesian product of  $N$  one-dimensional regions  $C_1(R) = [-R, R]$ . We know that the shaping gain of  $C_N(R)$  is unity, the same as the shaping gain of  $C_1(R)$ . Show this directly by calculating the volume and power of  $C_N(R)$ .
- 14-6. Suppose we transmit a spherically shaped  $N$ -dimensional lattice code with spectral efficiency  $\nu$  bits per complex symbol and power (variance)  $P$  per complex symbol. Suppose hypothetically that you can choose a lattice  $\Lambda_N$  for each  $N$  such that the fundamental volume  $V(\Lambda_N)$  stays constant.
- What is the increase in  $\nu$  when we go from  $N = 2$  to  $N = 4$  to  $N = 6$ ? (You can use the continuous approximation.)
  - What is the asymptotic increase in  $\nu$  between  $N = 2$  and  $N \rightarrow \infty$ ?
- 14-7. Let  $\mathbf{X}_K$  be a  $K$ -dimensional vector consisting of  $K$  components of an  $N$ -dimensional vector  $\mathbf{X}_N$ , the latter being a spherically uniform random vector. Show that for fixed  $K$ , as  $N \rightarrow \infty$ ,  $\mathbf{X}_K$  is a Gaussian vector with identically distributed independent components. **Hint:** Assume the radius is chosen so that each component of  $\mathbf{X}_N$  is normalized to unity variance. Then show that  $\mathbf{X}_K$  approaches a Gaussian density with unit-variance components.
- 14-8. Let  $\mathbf{X}$  be a spherically uniform random vector with radius  $R$  and dimension  $N$ . Show that for any  $0 < \epsilon \leq R$ ,

$$\Pr\{\|\mathbf{X}\| \leq R - \epsilon\} \rightarrow 0 \text{ as } N \rightarrow \infty. \quad (14.73)$$

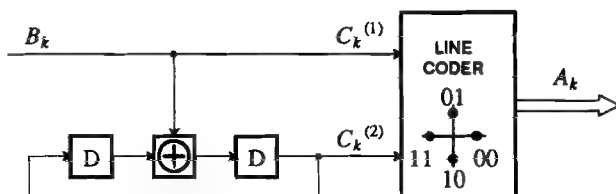
Thus, as  $N \rightarrow \infty$ , almost all the volume of a sphere is near its surface, and  $\|\mathbf{X}\|^2$  becomes  $R^2$  almost surely. An interpretation is that the multidimensional sphere is making maximum use of the available power by nearly always transmitting vectors that have this maximum power.

- 14-9. Design a 4-PSK line coder so that the slightly simpler trellis coder in Figure 14-31 has the same performance as the trellis coder in Figure 14-8. This convolutional coder (by itself, without the line coder) was shown in Problem 13-10 to be inferior to the one in Figure 14-8 in that the minimum Hamming distance is 3 instead of 5. However, with a properly designed line coder, the trellis code is not inferior. In fact it is equivalent to the coder in Figure 14-8. Another equivalent coder is studied in Problem 14-10.

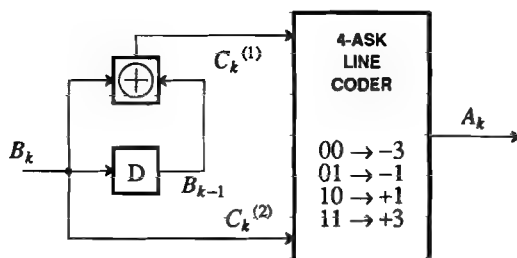


**Figure 14-31.** A four-state trellis coder with the same performance as that in Figure 14-8 can be made with a slightly simpler convolutional coder. The convolutional coder by itself is inferior in Hamming distance to the convolutional coder in Figure 14-8 (see Problem 13-10).

- 14-10. Show that the trellis coder in Figure 14-32 has the same performance as the one in Figure 14-8. Just as with Problem 14-9, the convolutional coder alone has  $d_{H,min} = 3$ , which is inferior to the  $d_{H,min} = 5$  of Figure 14-8. But the trellis coder is equally good.
- 14-11. Show that for the trellis coder in Figure 14-33 the distance of the minimum distance error event depends on the correct state trajectory. The code is nonlinear even though the convolutional coder used to make it is linear.
- 14-12. Suppose you are asked to design a 400 bit per second modem for a noisy bandlimited and power limited passband channel. You determine that with reasonable excess bandwidth, a symbol rate of 100 symbol per second is possible.
- For an uncoded system, what is the required alphabet size? Choose a constellation.
  - For a two-dimensional trellis coded system, what is the required alphabet size? Choose a constellation.
  - Suppose that to get an acceptable probability of error the uncoded system requires 4 dB more power than the channel can tolerate. How would you overcome this problem?
- 14-13. Find the distance of the second shortest error event for the trellis coder in Figure 14-10, assuming its line coder uses the 8-PSK constellation in Figure 14-13,  $|A_k| = a$ , and the trellis is defined by Figure 14-14.
- 14-14. Consider the  $M = 5$  cross constellation in Figure 6-28. Assume that symbols are of the form  $[a_1, a_2]$  where  $a_i \in \{\pm 5, \pm 3, \pm 1\}$ .
- Do set partitioning and determine the minimum distance between symbols in the set for all levels of partitioning, as in Figure 14-16.
  - What is the difference (in dB) in average power between this 32-cross constellation and a 16-QAM constellation where symbols are of the form  $[a_1, a_2]$  where  $a_i \in \{\pm 3, \pm 1\}$ ?



**Figure 14-32.** A four state trellis coder made with a feedback-type convolutional coder. This coder is discussed in Problem 14-10.

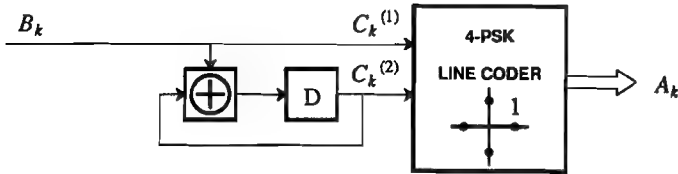


**Figure 14-33.** A nonlinear trellis code made using a linear convolutional coder. The minimum-distance error event depends on the correct state trajectory.

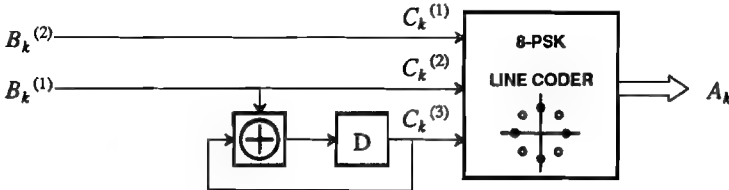
- (c) Suppose that you need a total gain of about 3 dB compared to a 16-QAM uncoded system. What is the minimum value of  $\bar{m}$  (the number of coded bits) that you could select for the code? Roughly how many states should the convolutional coder have?
- (d) Repeat part (c) assuming that a total gain of about 5 dB is required.
- 14-15.** Consider the coder in Example 14-15. Assume a convolutional coder equivalent to that in Figure 14-8.
- (a) Assuming that the parallel transitions form the minimum-distance error events, find the total gain of the trellis coder. **Hint:** Compare to an 8-PSK uncoded system.
- (b) Assign subsets from the third row of Figure 14-16 to transitions in the trellis, and find the distance of a length three error event, assuming the correct state trajectory is all zero. Does the assumption in part (a) look reasonable for this coder?
- 14-16.** Show that  $R$  defined by (9.150) is unbounded for the code in Example 14-17.
- 14-17.** Consider the trellis coder in Figure 14-34. The mapping is given by

$C_k^{(1)}$	$C_k^{(2)}$	$A_k$
0	0	+1
0	1	+j
1	0	-1
1	1	-j

- (a) Draw the state transition diagram and trellis with each arc labeled with the pair  $(B_k, A_k)$ .
- (b) Find the minimum-distance error events and their distance, and estimate their probability of occurring. Assume the channel adds white Gaussian noise with variance  $\sigma^2$ .
- (c) Compare this coded system with an uncoded 2-PSK system.
- (d) Consider the related system shown in Figure 14-35. Use set partitioning to design the line coder.



**Figure 14-34.** A Trellis coder with a 4-PSK output alphabet.



**Figure 14-35.** A trellis coder based on the one in Figure 14-34 but using an 8-PSK output alphabet and parallel state transitions.

- (e) Estimate the total gain at high SNR (compare to uncoded 4-PSK).
- 14-18. Consider the convolutional coder with feedback shown in Figure 14-36. Use set partitioning to design a mapping so that this trellis code performs as well as the 8-PSK trellis code with trellis shown in Figure 14-14.

## REFERENCES

1. R. de Buda, "Some Optimal Codes Have Structure," *IEEE Journal on Selected Areas in Communications*, p. 877 (Aug. 1989).
2. G. D. Forney, Jr and L-F Wei, "Multidimensional Constellations, Part I: Introduction, Figures of Merit, and Generalized Cross Constellations," *IEEE Journal on Selected Areas in Communications*, p. 877 (Aug. 1989).
3. G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Information Theory* IT-28, No. 1 (Jan. 1982).
4. L-F. Wei, "Rotationally Invariant Convolutional Channel Coding with Expanded Signal Space: Part II: Nonlinear Codes," *IEEE Journal on Selected Areas in Communications* SAC-2(5) (Sep. 1984).
5. G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets Part I: Introduction," *IEEE Communications Magazine* 25(2) pp. 5-11 (Feb. 1987).
6. G. D. Forney, Jr, "Coset Codes - Part I: Introduction and Geometrical Classification," *IEEE Trans. Information Theory* IT-34 p. 1123 (1988).
7. G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets Part II: State of the Art," *IEEE Communications Magazine* 25(2) pp. 12-21 (Feb. 1987).
8. A. R. Calderbank and N. J. A. Sloane, "Four-Dimensional Modulation With an Eight-State Trellis Code," *AT&T Technical Journal* 64(5) (May-June 1985).
9. G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient Modulation for Band-Limited Channels," *IEEE Journal on Selected Areas in Communications* SAC-2(5) (Sep. 1984).
10. L-F. Wei, "Trellis-Coded Modulation with Multi-Dimensional Constellations," *IEEE Trans. on Information Theory* IT-33 p. 483 (1987).
11. A. R. Calderbank and N. J. A. Sloane, "New Trellis Codes Based on Lattices and Cosets," *IEEE Trans. Information Theory* IT-33 p. 177 (1987).
12. M. V. Eyuboglu and G. D. Forney, Jr, "Trellis Precoding: Combined Coding, Precoding, and Shaping for Intersymbol Interference Channels," *IEEE Trans. Information Theory*, (March 1988).

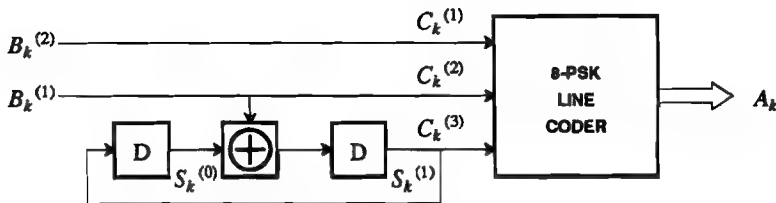


Figure 14-36. A convolutional coder with feedback, studied in Problem 14-18.

1992).

13. A. Duel-Hallen and C. Heegard, "Delayed Decision-Feedback Equalization," *IEEE Trans. Communications* COM-37 p. 13 (Jan. 1988).
14. P. Chevillat and E. Eleftheriou, "Decoding of Trellis-Encoded Signals in the Presence of Inter-symbol Interference and Noise," *IEEE Trans. Communications* COM-37 p. 669 (July 1989).
15. M. V. Eyuboglu and S. U. Qureshi, "Reduced-State Sequence Estimation for Coded Modulation on Intersymbol Interference Channels," *IEEE Journal Select Areas in Communications* SAC-7 p. 989 (Aug. 1989).
16. I. P. Csajka and G. Ungerboeck, "Method and Arrangement for Coding Binary Signals and Modulating a Carrier Signal," *U. S. Patent no. 4,077,021*, (Feb. 28, 1978).
17. E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications*, Macmillan, New York (1991).
18. A. R. Calderbank and J. Mazo, "A New Description of Trellis Codes," *IEEE Trans. on Information Theory* IT-30(6)(Nov. 1984).
19. D. Divsalar, M. K. Simon, and J. H. Yuen, "Trellis Coding with Asymmetric Modulations," *IEEE Trans. on Communications* COM-35(2)(Feb. 1987).
20. G. D. Forney, Jr and M. V. Eyuboglu, "Combined Equalization and Coding Using Precoding," *IEEE Communications Magazine*, (Dec. 1991).
21. M. V. Eyuboglu and G. D. Forney, Jr, "Advanced Modulation Techniques for V.Fast," *European Transactions on Telecommunications and Related Technologies*, (to appear).
22. G. Lang and F. Longstaff, "A Leech Lattice Modem," *IEEE Journal on Selected Areas in Communications* 7 p. 968 (Aug. 1989).
23. P. H. Siegel and J. K. Wolf, "Modulation and Coding for Information Storage," *IEEE Communications Magazine*, (Dec. 1991).

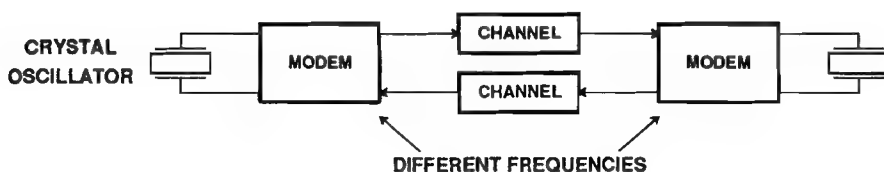
# 15

---

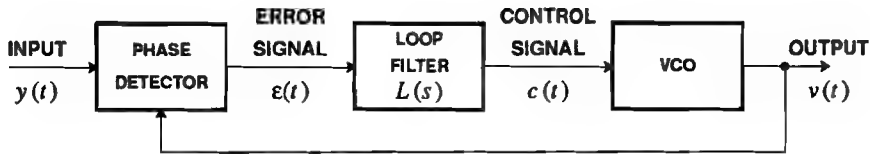
## PHASE-LOCKED LOOPS

---

In a continuous-time world, establishing a common time base at physically separated locations presents some serious challenges. Typical systems use independent time bases, frequently derived from crystal oscillators, as shown in Figure 15-1. Although crystal oscillators provide extremely accurate timing references at low cost, "extremely accurate" is not adequate to maintain the integrity of discrete-time data. Timing references often have to be identical, at least in the sense of long term averages. In other words, systems must be *synchronized*. Underlying most synchronization techniques is the phase-locked loop (PLL). In this chapter we derive the basic principles of PLLs. Two practical applications, carrier and timing recovery, are treated in-depth in Chapters 16 and 17.



**Figure 15-1.** Typical systems use independent time bases, frequently derived from crystal oscillators, at physically separated locations.



**Figure 15-2.** Basic structure of a continuous-time PLL. The VCO produces a signal or clock that tracks the phase of the input signal.

The basic PLL structure is shown in Figure 15-2. The *voltage-controlled oscillator* (VCO) attempts to produce a signal  $v(t)$  that tracks the phase of the input  $y(t)$ . A *phase detector* measures the phase error between the input  $y(t)$  and the VCO output  $v(t)$ . The resulting error signal can be filtered to become a *control signal* that drives the VCO. The basic idea is obvious—if the VCO phase gets ahead of the phase of the input, the control signal should be reduced. If the VCO phase gets behind, the control signal should be increased. As with any feedback system, the parameters must be chosen to ensure stability.

The goal in design of the PLL varies with the application.

#### Example 15-1.

In timing recovery (Chapter 17) on a single point-to-point link, such as in a voiceband data modem, the objective is to generate a stable single-frequency tone at the output of the VCO. The frequency of this tone should equal the average symbol rate of the input, but transient variations in the symbol rate should be ignored, as should noise or other interference. In fact, any fluctuations in the symbol rate detected by the timing recovery can be assumed to be a consequence of interference such as noise, because there is no mechanism in most channels for introducing significant fluctuations in the symbol rate. □

#### Example 15-2.

In carrier recovery (Chapter 16), by contrast, the objective is to track the phase of the carrier on the input signal as closely as possible, while at the same time minimizing the effect of noise. Unlike timing phase, several important channels can introduce significant fluctuations in the carrier phase and frequency. To properly demodulate the signal, these fluctuations should be replicated on the carrier used by the receiver for demodulation. The output of the VCO is therefore not a single-frequency tone (unless the phase of the carrier on the input does not vary). □

Other applications usually fall into one of these two categories as well. The objective is either a single-frequency, or closely-tracked phase, or a compromise between the two.

In practice, many PLLs look very different from that shown in Figure 15-2. There may be no explicit VCO, or the VCO may be built using digital circuitry arranged as a controllable countdown chain. The phase detector can be very complicated, sometimes actually consisting of an entire receiver, complete with adaptive equalizer, or very simple, consisting of an exclusive-or gate. A PLL may be implemented completely or partly in discrete-time, and completely or partly with digital



circuits. Although the relationship between the model in Figure 15-2 and an actual implementation may be subtle, the basic principles are the same for all implementations.

We will concentrate on the steady-state, in-lock behavior of the PLL, using only linearized analysis, ignoring important issues such as acquisition and non-linear behavior.

## 15.1. IDEAL CONTINUOUS-TIME PLL

PLLs are conceptually simple, but they are inherently non-linear systems and their analysis can be difficult. However, with some carefully crafted simplifying assumptions we can develop some powerful analytical tools that simplify the analysis.

### 15.1.1. Assumptions

First assume a particular form for the input

$$y(t) = A_y \cos(\omega_y t + \theta(t)), \quad (15.1)$$

where  $A_y$  and  $\omega_y$  are constants. Of course in practice the input is likely to be more complicated, having amplitude variations in addition to phase and frequency, for example, but as long as the design of the phase detector is appropriate for the form of a particular input, our analysis will be valid. The output of the VCO is assumed to have a similar form

$$v(t) = A_v \cos(\omega_v t + \phi(t)). \quad (15.2)$$

When  $\phi(t)$  is a constant the frequency of the VCO output is  $\omega_v$ , called the *natural* or *free-running frequency* of the VCO. It is for convenience that we express the input (15.1) in terms of the natural frequency of the VCO.

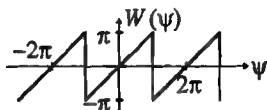
### 15.1.2. The Ideal Phase Detector

Assuming forms (15.1) and (15.2) the output of an ideal phase detector is

$$e(t) = W(\theta(t) - \phi(t)) \quad (15.3)$$

where the function  $W(\cdot)$ , shown in Figure 15-3, reflects the  $2\pi$  ambiguity in the phase difference. Because of the shape of  $W(\cdot)$ , this phase detector is called a *sawtooth phase detector*. We have assumed unity slope for the function  $W(\cdot)$ , although in practice the phase detector may exhibit some other gain, often written  $K_p$ . That gain is easily modeled as part of the loop filter gain, so its explicit inclusion is not necessary. Because of the  $2\pi$  ambiguity in an ideal phase detector, sudden changes of  $2\pi$  in  $\theta(t)$  or  $\phi(t)$  have no effect on the system (they are not detected by the phase detector). Such changes are called *clicks*, and are usually detrimental.

We will see many variations of this basic phase detector. It is also often possible to design *frequency detectors* that do not suffer this  $2\pi$  phase ambiguity [1].



**Figure 15-3.** An ideal phase detector can only detect phase errors  $\psi$  modulo  $2\pi$ . This is equivalent to applying this function  $W(\cdot)$  to the phase error  $\psi$ . Because of the shape of  $W(\cdot)$ , this phase detector is called a sawtooth phase detector.

### 15.1.3. The Ideal VCO

The ideal VCO, with properties summarized in Figure 15-4, produces the output (15.2), which has instantaneous frequency

$$\frac{d}{dt}[\omega_v t + \phi(t)] = \omega_v + \frac{d\phi(t)}{dt}. \quad (15.4)$$

Again, a practical VCO may have gain, often written  $K_v$ , that can be modeled as part of the gain of the loop filter. Intuitively, we would like to directly control the instantaneous frequency with the control input  $c(t)$ . The VCO should therefore be designed so that

$$\frac{d\phi(t)}{dt} = c(t). \quad (15.5)$$

#### Example 15-3.

A constant control signal  $c(t) = K$  will produce the constant frequency  $\omega_v + K$  at the output.  $\square$

It is sometimes convenient for analysis to take the Laplace transform of (15.5),

$$s\Phi(s) = C(s) = L(s)E(s), \quad (15.6)$$

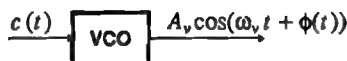
where  $C(s)$  is the Laplace transform of the control signal and  $E(s)$  is the Laplace transform of the error signal  $\epsilon(t)$ .

### 15.1.4. Phase and Average-Frequency Lock

The ideal PLL is *phase locked* if

$$\phi(t) = \theta(t) + \phi \quad (15.7)$$

for some constant  $\phi$ . If  $\phi = 0$ , the PLL is *perfectly* phase locked. In other words, the



**Figure 15-4.** An ideal VCO is summarized in this figure. The instantaneous frequency of the output will be  $\omega_v + c(t)$ .

VCO output is exactly tracking the phase of the input. It is locked to an average frequency  $\omega_v + K$  if

$$\phi(t) = Kt \quad (15.8)$$

for some constant  $K$ . The VCO output frequency is presumably exactly the same as the input average frequency. In Example 15-1 it is more important to have average-frequency lock than phase lock, while in Example 15-2 the situation is reversed.

Intuitively, there must be some limitations on the input phase  $\theta(t)$  for the PLL to be phase or average-frequency locked because the phase detector output is bounded by  $\pm \pi$ . To find the limitations, assume a simple form for the phase of the input,

$$\theta(t) = \omega_0 t + \theta. \quad (15.9)$$

In other words, the input  $y(t)$  is a sinusoid with frequency  $\omega_v + \omega_0$  and phase  $\theta$ , a constant. Assume the PLL is phase locked. In order for it to remain phase locked, the *frequency offset*  $\omega_0$  must not exceed a limited range called the *lock range* or *hold-in range* of the PLL. It is easy to derive the lock range.

#### Exercise 15-1.

Assuming that the input phase has the form (15.9), show that the PLL can only maintain phase lock if

$$|\omega_0| \leq \pi |L(0)|, \quad (15.10)$$

where  $L(0)$  is the d.c. gain of the loop filter (the Laplace transform evaluated at  $s = 0$ ). Assume an ideal phase detector and VCO.  $\square$

### 15.1.5. Analysis of the Linearized Dynamics

Phase and average-frequency lock are static concepts—they assume the PLL is in steady state. If we assume that the phase error is small enough for all  $t$ ,

$$|\theta(t) - \phi(t)| < \pi \quad (15.11)$$

then the phase detector is operating in its *linear range* (see Figure 15-3),

$$\varepsilon(t) = \theta(t) - \phi(t). \quad (15.12)$$

and the analysis of the dynamics of the PLL is simple. The transfer function from the phase  $\theta(t)$  of the input to the phase  $\phi(t)$  of the VCO follows by taking the Laplace transform of (15.12),

$$E(s) = \Theta(s) - \Phi(s), \quad (15.13)$$

and from (15.6),

$$E(s) = \frac{s\Phi(s)}{L(s)}. \quad (15.14)$$

Combining these and solving for  $\Phi(s)/\Theta(s)$  we get the *phase transfer function*

$$\frac{\Phi(s)}{\Theta(s)} = \frac{L(s)}{L(s) + s}. \quad (15.15)$$

The phase transfer function summarizes many of the important features of the PLL.

### Exercise 15-2.

Assume that  $L(s) = N(s)/D(s)$  is a rational Laplace transform where the degree of  $N(s)$  (the number of zeros) is less than or equal to the degree of  $D(s)$  (the number of poles). Show that the number of poles in  $\Phi(s)/\Theta(s)$  (called the *order* of the PLL) is one plus the number of poles in the loop filter  $L(s)$ .  $\square$

### Example 15-4.

A *first-order PLL* is characterized by having the simple loop filter

$$L(s) = K_L. \quad (15.16)$$

In this case, the transfer function can be written

$$\frac{\Phi(s)}{\Theta(s)} = \frac{K_L}{K_L + s}. \quad (15.17)$$

This is a single-pole lowpass filter with its pole at  $s = -K_L$ . It is stable as long as  $K_L > 0$ , and its lock range is found from (15.10) to be

$$|\omega_0| \leq \pi |K_L|. \quad (15.18)$$

Its lowpass characteristic is a very useful property for many applications. When the application requires average-frequency lock, as in Example 15-1, then a narrowband lowpass PLL will be useful. Even when phase tracking is required, as in Example 15-2, a lowpass PLL can help reject some of the noise, particularly if it is known that the phase variations that we wish to track are relatively slow.  $\square$

Evaluating transfer function (15.15) at  $s = 0$ , the PLL has unity gain for d.c. phase errors. In other words, when the input phase is constant,  $\theta(t) = K$ , then the output phase is the same constant,  $\phi(t) = K$ . In this case we get perfect phase lock with any loop filter.

### Exercise 15-3.

Show that if the input has frequency offset,  $\theta(t) = \omega_0 t + K$ ,  $\omega_0 \neq 0$ , then the first-order PLL of Example 15-4 cannot achieve perfect phase lock. It can achieve phase lock if  $\omega_0$  is within the lock range, but there will always remain a phase error.  $\square$

The *bandwidth* of a PLL is loosely defined to be the bandwidth of the transfer function  $\Phi(s)/\Theta(s)$ . Lowering the bandwidth means increasing the attenuation of high frequency components in the input phase or noise, but for the first order PLL, it also reduces the lock range. It is possible to reduce the bandwidth *without* reducing the lock range by using a second-order PLL.

### Example 15-5.

Consider a *type I second-order PLL* with loop filter

$$L(s) = K_L \frac{s + K_1}{s + K_2}. \quad (15.19)$$

From Exercise 15-1 the lock range is

$$|\omega_0| \leq \pi |K_L K_1 / K_2| . \quad (15.20)$$

From (15.15) the phase transfer function is

$$\frac{\Phi(s)}{\Theta(s)} = \frac{K_L s + K_L K_1}{s^2 + (K_L + K_2)s + K_L K_1} . \quad (15.21)$$

It can be shown that this is stable as long as  $K_2 > -K_L$  and  $K_L K_1 > 0$  (see Problem 15-3).

□

### Example 15-6.

Suppose that

$$L(s) = \frac{s + 1}{s - 0.5} . \quad (15.22)$$

The loop filter itself is not stable, but the closed-loop PLL is. The transfer function is

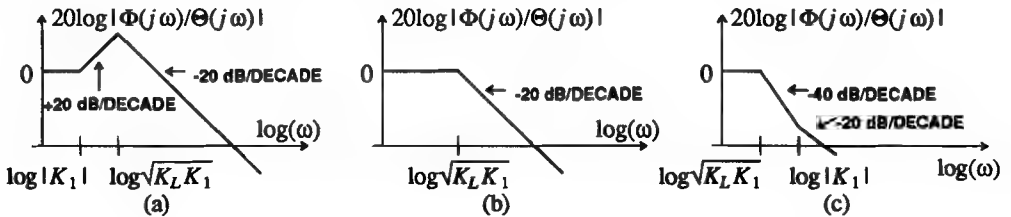
$$\frac{\Phi(s)}{\Theta(s)} = \frac{s + 1}{s^2 + 0.5s + 1} \quad (15.23)$$

which has poles at

$$-0.25 \pm j0.97 , \quad (15.24)$$

both in the left half plane. □

Since the transfer function of a second-order PLL has a zero and two poles, the rolloff at high frequencies is the same (20 dB/decade) as for the first-order PLL. Three possible Bode plots of the phase transfer function are shown in Figure 15-5. The bandwidth of the PLL is determined primarily by  $K_L K_1$ . But the lock range (15.20) is determined by both  $K_L K_1$  and  $K_2$ , as shown in (15.20). In principle, we can make the lock range as large as we like by decreasing  $K_2$  while keeping the bandwidth constant by keeping  $K_L K_1$  constant. This is the main advantage of the type I second-order PLL. However, there is potentially one serious problem. Although the gain at d.c. is always unity (evaluate (15.21) at  $s = 0$ ), the closed-loop gain may be greater than



**Figure 15-5.** Three possible Bode plots of the phase transfer function (15.21) of the type I second-order PLL, assuming the poles are complex. If  $|z|$  is the magnitude of the zero and  $|p|$  is the magnitude of the poles, then in (a)  $|z| < |p|$ , in (b)  $|z| = |p|$ , and in (c)  $|z| > |p|$ . Note that the bandwidth of the PLL is independent of  $K_2$ .

unity in some range of frequencies. In this case, some components of the phase of the input will be *amplified*, which is often not desired. This phenomenon is known as *peaking*.

#### Example 15-7.

In Example 15-6,  $K_1 = K_L = 1$ , so the bode plot is given by Figure 15-5b. The magnitude response is

$$\left| \frac{\Phi(j\omega)}{\Theta(j\omega)} \right|^2 = \left| \frac{j\omega + 1}{-\omega^2 + 0.5j\omega + 1} \right|^2. \quad (15.25)$$

If we evaluate this at the natural frequency  $\omega_n = 1$  (the natural frequency is equal to the magnitude of the poles), we find that

$$\left| \frac{\Phi(j)}{\Theta(j)} \right|^2 = \left| \frac{j+1}{0.5j} \right|^2 = 8. \quad (15.26)$$

If the input phase  $\theta(t)$  has a component at  $\omega_n = 1$ , then the output phase  $\phi(t)$  will have that same component eight times larger! The frequency response is sketched in Figure 15-6.  $\square$

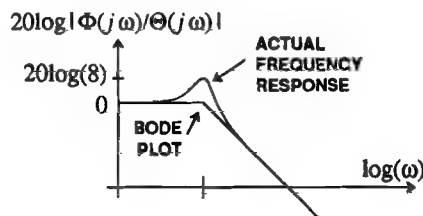
Peaking is not always a serious impairment, but sometimes it is devastating.

#### Example 15-8.

In some systems, such as long transmission lines with many repeaters (Chapter 1), many PLLs are cascaded in series. If the PLLs have greater than unity gain for a phase at the input that varies at any particular frequency, the amplification of that phase can become quite severe after just a few PLLs. This issue is explored in Chapter 17 for the specific example of timing recovery PLLs in the repeaters.  $\square$

The peaking properties of type I second-order PLLs are studied in detail in Problem 15-6. In particular, it is shown that if peaking is disallowed, the lock range of a second-order PLL is actually *smaller* than the lock range of a first-order PLL with comparable bandwidth. Hence, if peaking cannot be tolerated then the primary advantage of second-order loops evaporates.

The following *type II second-order PLL* is a special case of the type I PLL when  $K_2 = 0$ ,



**Figure 15-6.** The frequency response of the type I second-order PLL in Example 15-6 exhibits peaking in which the phase at certain frequencies is amplified by the PLL.

$$L(s) = K_L \frac{s + K_1}{s} . \quad (15.27)$$

This is sometimes called a *proportional plus integral* loop filter. From (15.21) the closed-loop phase response is

$$\frac{\Phi(s)}{\Theta(s)} = \frac{K_L K_1 + K_L s}{K_L K_1 + K_L s + s^2} . \quad (15.28)$$

As with the previous PLLs, this one has unity gain at d.c. Unlike the previous PLLs, however, it has an integrator in the loop filter. In fact, by convention, the "type" of a PLL is the number of integrators in the loop filter plus one. Its main advantage is that the integrator leads to perfect phase lock even in the face of frequency offset. A disadvantage is that it always exhibits peaking (see Problem 15-7). Two other second-order PLL loop filters are studied in Problem 15-1 and Problem 15-8.

### 15.1.6. Steady-State Response

It is often useful to know precisely the steady-state operating point of a PLL given certain inputs. The steady-state phase error is defined to be

$$\epsilon_{ss} = \lim_{t \rightarrow \infty} \epsilon(t) . \quad (15.29)$$

If the PLL does not achieve perfect phase lock then  $\epsilon_{ss} \neq 0$ . If  $\epsilon(t) = 0$  for  $t < 0$  then we can (usually) find  $\epsilon_{ss}$  using the *final value theorem* for Laplace transforms,

$$\epsilon_{ss} = \lim_{s \rightarrow 0} sE(s) . \quad (15.30)$$

Combining (15.14) and (15.15) we get the Laplace transform of  $\epsilon(t)$  in terms of the input phase,

$$E(s) = \frac{s\Theta(s)}{L(s) + s} \quad \epsilon_{ss} = \lim_{s \rightarrow 0} \frac{s^2\Theta(s)}{L(s) + s} . \quad (15.31)$$

#### Example 15-9.

Suppose the input phase is

$$\theta(t) = \omega_0 t u(t) \quad \Theta(s) = \omega_0 / s^2 \quad (15.32)$$

where  $u(t)$  is the unit step. In other words, at time  $t = 0$  the input suddenly acquires a frequency offset of  $\omega_0$ . There will of course be transients in the response of the PLL, but after the transients die out, from (15.31) the steady-state phase error will be

$$\epsilon_{ss} = \lim_{s \rightarrow 0} \frac{\omega_0}{L(s) + s} . \quad (15.33)$$

For the first-order PLL,  $L(s) = K_L$  and

$$\epsilon_{ss} = \frac{\omega_0}{K_L} . \quad (15.34)$$

The steady-state error is non-zero, but can be reduced by increasing the loop gain (at the expense of increasing the bandwidth of the loop, see Problem 15-2). For the type I second-

order PLL with loop filter given by (15.19),

$$\epsilon_{ss} = \frac{K_2 \omega_0}{K_1 K_L}. \quad (15.35)$$

The steady-state error is again non-zero, but can be reduced this time by making  $K_2$  small. However, this may lead to peaking, or greater than unity gain for inputs at some frequencies (see Problem 15-6). If  $K_2 = 0$ , we have the type II loop of (15.27) which has zero steady-state error,  $\epsilon_{ss} = 0$ . This is the main advantage of type II second-order PLLs. Intuitively, the integrator in the loop filter holds a constant at its output proportional to the frequency offset. As shown in Problem 15-7, there is always peaking in this PLL, but the peaking can be made small by making  $K_1$  small.  $\square$

At least one integrator in the loop filter is required to get zero steady-state error in the face of frequency offset.

### 15.1.7. Transients

The previous analysis assumes that the phase error is always small, so that the phase detector operates in its linear range. When the PLL is locked to an input signal, this is a reasonable assumption. But during *capture*, it is not. Consider a PLL that is locked to an input signal with frequency equal to the natural frequency  $\omega_n$  of the VCO. Then the control signal  $c(t)$  is zero. Suppose that suddenly the frequency of the input changes. Because of the loop filter and practical limitations of the VCO, the VCO will not respond instantly to lock onto the new frequency. As a consequence, for a period of time the phase difference between the input and VCO output can get large, and in-fact can easily slip cycles (making sudden jumps of magnitude  $2\pi$  for the ideal phase detector). During this time, the linearized analysis is not valid.

The lock range is the range of input frequencies over which an in-lock PLL will stay locked. But even within this range the PLL may not be able to capture the input frequency if it starts out unlocked. The *capture range* is defined as the range of input frequencies over which an initially unlocked PLL can lock. The *pull-in* time is the amount of time it takes the PLL to lock. A subset of the capture range into which the PLL can lock without cycle slipping also has a name, the *seize* range. Complete analysis of these effects is complicated by the non-linear nature of the PLL, and is left to references in Section 15.5.

## 15.2. DISCRETE-TIME PLLs

In digital communications systems, completely analog continuous-time PLLs like those studied in Section 15.1 are rare. Most are hybrid analog/digital or mixed continuous and discrete-time. We will begin with the discrete-time PLL. Mixed systems are considered in Sections 15.3 and 15.4 where alternative phase detector and VCO designs are discussed.



### 15.2.1. The Basic Model

A discrete-time PLL is shown in Figure 15-7. Assumptions about the form of the input signal and the output of the VCO are shown in the figure, and are analogous to those for the continuous-time PLL. The phase detector is a discrete-time version of that considered before, and has output

$$\varepsilon_k = W(\theta_k - \phi_k) \quad (15.36)$$

where  $W(\cdot)$  is shown in Figure 15-3.

The discrete-time VCO, although analogous to the continuous-time VCO, is not quite as obvious. Analogous to differential equation (15.5), the phase  $\phi_k$  satisfies the difference equation

$$\phi_{k+1} - \phi_k = c_k. \quad (15.37)$$

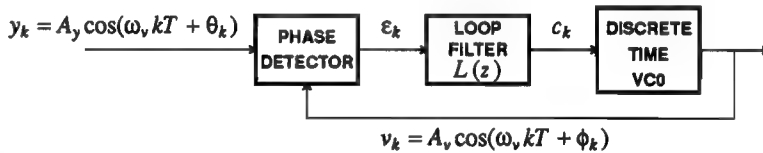
Using this, the output of the VCO can be written

$$\begin{aligned} v_{k+1} &= A_v \cos(\omega_v(k+1)T + \phi_{k+1}) \\ &= A_v \cos(\omega_v kT + \phi_k + \omega_v T + c_k). \end{aligned} \quad (15.38)$$

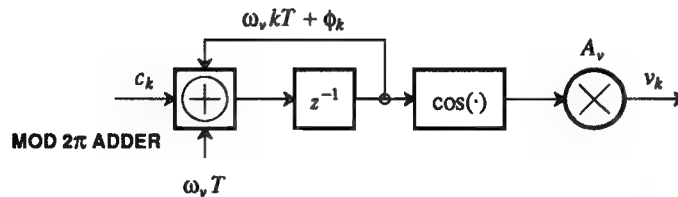
This leads to the structure in Figure 15-8. Taking the Z transform of (15.37) we get

$$\Phi(z) = \frac{1}{z-1} C(z) = \frac{L(z)}{z-1} E(z) \quad (15.39)$$

where  $L(z)$  is the loop filter transfer function and  $E(z)$  is the Z transform of the error



**Figure 15-7.** A discrete-time PLL with assumptions about the form of the input and the output shown.



**Figure 15-8.** A discrete-time VCO consists of an accumulator and cosine computation, along with some constant additions and multiplications. The modulo-two adders reflect the fact that the numbers being added are angles (in radians).

signal  $\epsilon_k$ .

#### Exercise 15-4.

Assume the input has frequency offset  $\omega_0$

$$\theta_k = \omega_0 kT + \theta \quad (15.40)$$

where  $\theta$  is some constant and  $T$  is the sample interval. Show that the lock range is

$$|\omega_0| \leq \frac{\pi}{T} |L(1)|. \quad (15.41)$$

□

### 15.2.2. Analysis of the Dynamics

As before, to analyze the dynamics we assume that the phase error is small enough that the phase detector is linear. The phase detector output is

$$\epsilon_k = \theta_k - \phi_k \quad (15.42)$$

or taking Z transforms

$$E(z) = \Theta(z) - \Phi(z). \quad (15.43)$$

Combining (15.43) with (15.39) we get the phase transfer function of the PLL,

$$\frac{\Phi(z)}{\Theta(z)} = \frac{L(z)}{L(z) + z - 1}. \quad (15.44)$$

By evaluating this at  $z = 1$  we see that just as with the continuous-time PLL, discrete-time PLLs have unity gain to d.c. phase inputs.

#### Example 15-10.

A first-order discrete-time PLL has loop filter  $L(z) = K_L$ , so

$$\frac{\Phi(z)}{\Theta(z)} = \frac{K_L}{K_L + z - 1}. \quad (15.45)$$

This has a pole at  $z = 1 - K_L$  and hence is stable if and only if  $0 < K_L < 2$ . Unlike the continuous-time PLL, there is an upper limit on the loop gain imposed by the stability requirement. Also, the phase transfer function is only a lowpass filter if  $0 < K_L < 1$  (see Problem 15-11). In fact, if  $1 < K_L < 2$  then inputs at all frequencies are amplified, which is probably not what we had hoped for! □

#### Example 15-11.

A general second-order discrete-time PLL has loop filter

$$L(z) = \frac{a_1 z + a_0}{b_1 z + b_0}. \quad (15.46)$$

The transfer function to phase is

$$\frac{\Phi(z)}{\Theta(z)} = \frac{a_1 z + a_0}{b_1 z^2 + (a_1 + b_0 - b_1)z + (a_0 - b_0)}. \quad (15.47)$$

□

### 15.2.3. Steady-State Error

Just as with continuous-time PLLs, the steady-state error is

$$\epsilon_{ss} = \lim_{k \rightarrow \infty} \epsilon_k . \quad (15.48)$$

If  $\epsilon_k = 0$  for  $k < 0$  we can (usually) use the final value theorem for Z transforms to write

$$\epsilon_{ss} = \lim_{z \rightarrow 1} (z - 1)E(z) . \quad (15.49)$$

Combining (15.43) and (15.44) we get an expression for  $E(z)$ ,

$$E(z) = \frac{\Theta(z)(z - 1)}{L(z) + z - 1} . \quad (15.50)$$

#### Example 15-12.

Suppose that the input frequency is exactly the natural frequency of the VCO, but a phase offset is introduced at time  $k = 0$ ,

$$\theta_k = \theta u_k , \quad (15.51)$$

where  $\theta$  is a constant and  $u_k$  is the unit step. The Z transform is

$$\Theta(z) = \frac{z\theta}{z - 1} . \quad (15.52)$$

Hence

$$\epsilon_{ss} = \lim_{z \rightarrow 1} \frac{(z - 1)z\theta}{L(z) + z - 1} . \quad (15.53)$$

For any  $L(z)$  such that  $L(1) \neq 0$ ,  $\epsilon_{ss} = 0$ , and the phase error decays to zero. □

#### Example 15-13.

Suppose that the input has frequency offset introduced at time  $k = 0$ ,

$$\theta_k = \omega_0 k u_k . \quad (15.54)$$

The Z transform is

$$\Theta(z) = \frac{z\omega_0}{(z - 1)^2} . \quad (15.55)$$

Hence

$$\epsilon_{ss} = \lim_{z \rightarrow 1} \frac{z\omega_0}{L(z) + z - 1} , \quad (15.56)$$

which will be zero if and only if  $L(z)$  has a pole at  $z = 1$ . Thus to have perfect phase lock in the face of frequency offset, the discrete-time loop filter must have a pole at  $z = 1$ , just as the continuous-time loop filter has to have a pole at  $s = 0$ . □

The "type" of a discrete-time PLL is defined to be one plus the number of poles at

$z = 1$ . From the previous example we see that a type II PLL has  $\epsilon_{ss} = 0$  when there is frequency offset.

## 15.3. PHASE DETECTORS

So far we have assumed an ideal phase detector with the characteristic shown in Figure 15-3. As a result of its shape, this phase detector is called a *sawtooth* phase detector. A wide variety of other phase detectors are used, ranging from simple to complicated. Much of the design effort in carrier and timing recovery is in the design of the phase detector (Chapters 16 and 17). In this section we describe some variations on the basic sawtooth phase detector.

### 15.3.1. Sinusoidal Phase Detectors

Prevalent throughout the history of PLLs is the *sinusoidal phase detector*. In fact it is so prevalent that some books on PLLs never consider any other type. Its dominance is a consequence of its easy implementation in analog circuitry. The structure is shown in Figure 15-9.

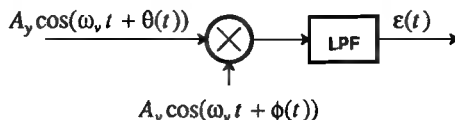
One significant difference between this phase detector and the sawtooth phase detector previously considered is that the VCO will tend to phase lock to the input *in quadrature* (with a 90 degree phase difference). In other words,  $\phi(t)$  in (15.2) will have a constant  $\pi/2$  term (or equivalently the  $\cos(\cdot)$  will be replaced with a  $\sin(\cdot)$ ). Assuming (15.1) and (15.2) are the input signal and the VCO output, the output of the multiplier is

$$\frac{A_v A_y}{2} \left[ \cos[\theta(t) - \phi(t)] + \cos[2\omega_v t + \theta(t) + \phi(t)] \right]. \quad (15.57)$$

Assuming the second term is removed by the LPF in Figure 15-9 we can write

$$\epsilon(t) = \frac{A_v A_y}{2} \cos[\theta(t) - \phi(t)]. \quad (15.58)$$

At first glance this does not look like a good estimate of the phase error  $\theta(t) - \phi(t)$ . In fact,  $\epsilon(t)$  is at its *maximum* when  $\theta(t) = \phi(t)$  (see Problem 15-14). Suppose that the PLL tries to minimize  $\epsilon(t)$  anyway. It will be minimized (in fact  $\epsilon(t) = 0$ ) when



**Figure 15-9.** A sinusoidal phase detector uses a multiplier and a lowpass filter. It is relatively easy to implement using analog circuits.

$$[\theta(t) - \phi(t)] \bmod 2\pi = \frac{\pi}{2}. \quad (15.59)$$

Thus, this PLL will minimize  $\epsilon(t)$  by maintaining a constant 90 degree ( $\pi/2$  radian) difference between the phase of the input and the phase of the VCO. Such a PLL is said to be phase locked in quadrature. Define

$$\psi(t) = \phi(t) + \frac{\pi}{2}. \quad (15.60)$$

Then the PLL will be in quadrature phase lock if  $\psi(t) = \theta(t)$ . The output of the VCO can be written

$$v(t) = A_v \cos(\omega_v t + \psi(t) - \frac{\pi}{2}) = A_v \sin(\omega_v t + \psi(t)). \quad (15.61)$$

This explicitly shows the 90 degree phase difference. We can now consider  $\psi(t)$  to be the phase of the VCO, and write the phase error

$$\epsilon(t) = \frac{A_v A_y}{2} \sin(\theta(t) - \psi(t)). \quad (15.62)$$

Now  $\epsilon(t)$  is a much more reasonable estimate of the phase error. In fact, the only difference now between this PLL and the continuous-time PLL of Section 15.1 is the function  $W(\cdot)$  which is now defined to be

$$W(x) = \frac{A_v A_y}{2} \sin(x), \quad (15.63)$$

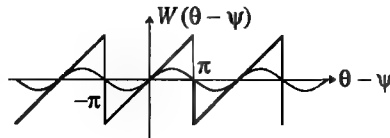
where  $x$  is the phase error  $\theta(t) - \psi(t)$ . This is plotted in Figure 15-10. The polarity of  $W(\cdot)$  is the same as that of the sawtooth phase detector for all  $x$ . For small phase errors  $x$ , the two phase detectors are very similar, and have approximately linear characteristics, since

$$\sin(x) \approx x, \quad (15.64)$$

so when  $\theta(t)$  is close to  $\psi(t)$ ,

$$\epsilon(t) \approx \frac{A_v A_y}{2} (\theta(t) - \psi(t)). \quad (15.65)$$

This differs from (15.12) only by a constant, so the analysis carried out in Section 3.1



**Figure 15-10.** A sinusoidal phase detector operates much like the sawtooth phase detector except for two things: the VCO output is in quadrature with the input, and the  $W(\cdot)$  function is sinusoidal, as shown in this figure. Also shown for reference is  $W(\cdot)$  for the sawtooth phase detector.

applies without modification if the phase error is assumed to be small.

One major practical difference between the sinusoidal phase detector and the ideal phase detector of Section 15.1 is the dependence of  $\epsilon(t)$  on the input signal level  $A_y$ , which in practice may be time-varying. This effect can be analyzed, as can the non-linearity that occurs for larger phase errors, but the analysis is much more difficult, and is beyond the scope of this book (see for example [2]).

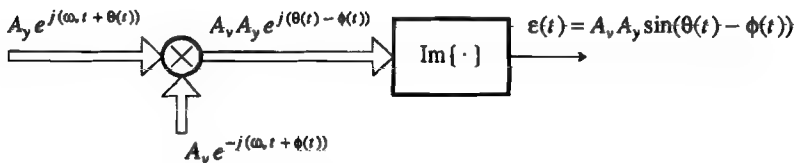
### 15.3.2. Complex Phase Detectors

For passband PAM systems it is common to do much of the signal processing on the complex-valued baseband equivalent signal, as shown in Chapter 6. A simple extension of the sinusoidal phase detector for complex signals is shown in Figure 15-11. For small phase errors, the phase detector is approximately linear,

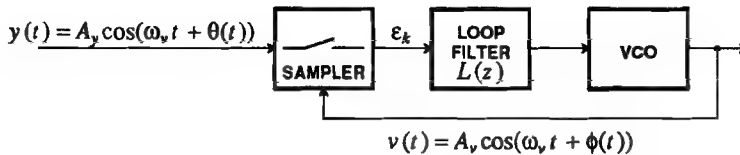
$$\epsilon(t) \approx A_v A_y [\theta(t) - \phi(t)] . \quad (15.66)$$

### 15.3.3. Sampling Phase Detectors

PLLs in digital communication systems are often a mixture of discrete and continuous-time subsystems. A common scenario is that the PLL determines when the incoming signal is sampled, and the samples are used to estimate the phase error. A simple system of this type is shown in Figure 15-12. The same forms for the input and VCO output are assumed (although in practical situations a digital VCO is more common, see Section 15.4 below). The sampling instants are at the upward-going zero crossings of the VCO output  $v(t)$ . Denote these sampling instants  $t = \tau_k$ , and note that the upward-going zero crossings of  $v(t)$  occur when  $\tau_k$  satisfies



**Figure 15-11.** A simple extension of the sinusoidal phase detector for complex signals is shown here.



**Figure 15-12.** A mixed discrete and continuous-time PLL with a sampling phase detector. The sampling of the input is directed by a continuous-time VCO.

$$\omega_v \tau_k + \phi(\tau_k) = k 2\pi - \frac{\pi}{2} . \quad (15.67)$$

Hence we can write

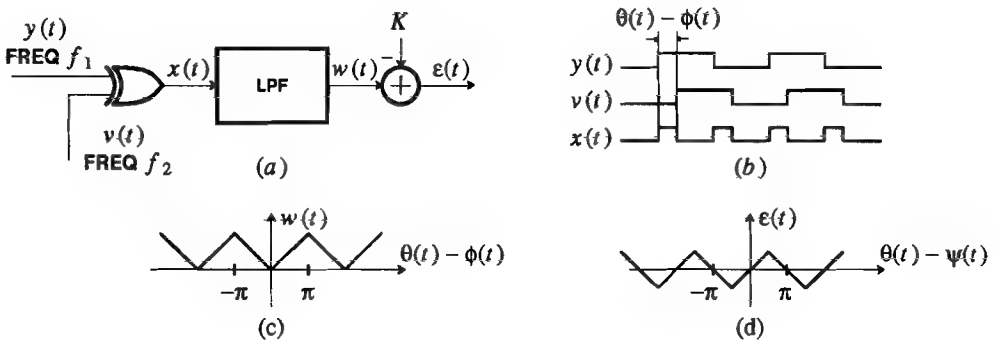
$$\begin{aligned} \epsilon_k = y(\tau_k) &= A_y \cos(k 2\pi - \frac{\pi}{2} - \phi(\tau_k) + \theta(\tau_k)) \\ &= A_y \sin(\theta(\tau_k) - \phi(\tau_k)) . \end{aligned} \quad (15.68)$$

Thus we *almost* have a discrete-time sinusoidal phase detector; the measured phase error  $\epsilon_k$  is *almost* a discrete-time version of (15.62). The reason we say "almost" is that since  $\tau_k$  is controlled by the VCO, the sampling times are non-uniform! However, we can often assume that  $\theta(t)$  and  $\phi(t)$  are varying slowly enough that the non-uniform sampling has little effect. With this approximation, the behavior of the PLL will be the same as that of discrete-time PLL with a sinusoidal phase detector.

The sampling phase detector in Figure 15-12 is commonly used for timing recovery (see Chapter 17). The PLL in Figure 15-12 is sometimes called a *digital PLL*, but this terminology is confusing because many other PLLs have digital elements. For a more detailed analysis, including the effect of the non-uniform sampling, see [3,4,5].

### 15.3.4. Exclusive-Or Phase Detectors

Another commonly used phase detector is the *exclusive-or phase detector*, illustrated in Figure 15-13 assuming that the inputs are square waves (digital continuous-time signals) rather than sinusoids (analog signals). In some applications the signals are square waves to begin with (see Section 15.4), but even when they are not, square waves are easily generated from sinusoidal signals using hard limiters. The output  $x(t)$  of the exclusive-or gate is high whenever the two input signals are different, as



**Figure 15-13.** An exclusive-or phase detector (a) assumes that the inputs are square waves (b) (digital continuous-time signals) rather than sinusoids (analog signals). The output of the lowpass filter (c) is proportional to the average amount of time that the two inputs signals differ in each cycle. After subtracting a constant, the result is a triangular phase detector characteristic (d).

shown in Figure 15-13b. The average duty cycle is proportional to the phase error. Thus if  $x(t)$  is lowpass filtered to perfectly extract the d.c. component while rejecting the fundamental and harmonics, the result is the function  $w(t)$  shown in Figure 15-13c. This is not directly useful because it is not linear near zero phase error. Subtracting a constant, the *triangular phase detector* characteristic illustrated in Figure 15-13d is obtained. If the constant  $K$  is correct, the VCO phase locks in quadrature, just like the sinusoidal phase detector.

When the input signal is not a square wave, but rather is sinusoidal, a square wave can be synthesized by hard limiting.

### 15.3.5. Phase Domain PLLs

For many application it is unnecessary to generate the VCO output explicitly. In other words, the sinusoid (or square wave) phase locked to the input is not needed, only its phase is needed. An example of a PLL that works entirely in the phase domain is shown in Figure 15-14. The phase of the input is measured with respect to a fixed clock, and that phase provides the input to the loop. We can call the frequency of the fixed reference clock  $\omega_v$ , and the phase measurement  $\theta_k$ . The phase measurement itself can be implemented digitally, for example by measuring the time between zero crossings in the reference and the input. The basic operation of this PLL is no different from those considered above.

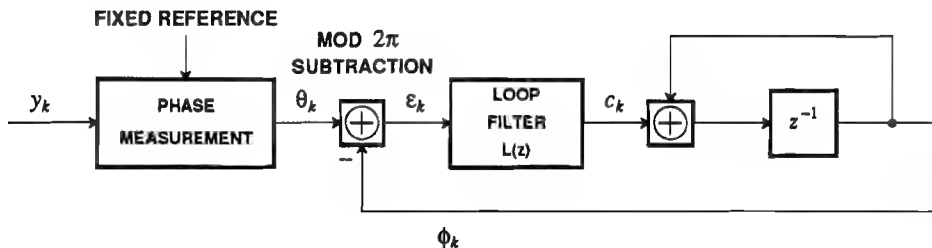
#### Exercise 15-5.

Show that the phase transfer function of the PLL in Figure 15-14 is the same as (15.44).  $\square$

An application of a phase domain PLL to the demodulation of PSK signals is given in Chapter 16.

### 15.3.6. Frequency Detectors

The phase detector is sufficient for maintaining phase lock for a PLL when the input frequency is within the lock range. We have also seen that the lock range is dependent on the loop bandwidth and the design of the phase detector. This leads to an undesirable tradeoff between the acquisition properties of a PLL and the in-lock



**Figure 15-14.** A PLL that works entirely in the phase domain. The phase of the input is measured with respect to a fixed clock, and that phase provides the input to the loop.



performance. For most designs these problems are manageable. However, there are cases where adequate lock range cannot be achieved while maintaining a sufficiently narrow loop bandwidth.

#### Example 15-14.

In a digital radio system (Section 5.4) the carrier frequency of the radio is determined by the free-running frequency of a microwave oscillator. The VCO in the receiver is also a microwave oscillator. Even if these oscillators have extremely accurate free-running frequencies, the offset can be large relative to the IF carrier frequency. These carrier frequency offsets are often on the order of 500 kHz or even larger. A PLL design which has a lock range this large would have much too large a closed-loop bandwidth to maintain adequately small phase jitter.  $\square$

In this instance, something must be done to increase the lock range without affecting the in-lock loop bandwidth. Two techniques are available for this purpose: *frequency sweeping* and the addition of a *frequency detector* to the phase detector.

A frequency detector is intuitively simple — it compares the frequency of the incoming signal to the local oscillator rather than comparing its phase. This is equivalent to measuring the phase without the  $2\pi$  ambiguity exhibited by phase detectors. See [1,6] for a description of frequency detectors. A class of phase detectors known as *adaptive phase detectors* can also function as frequency detectors and are described in [7,8,9].

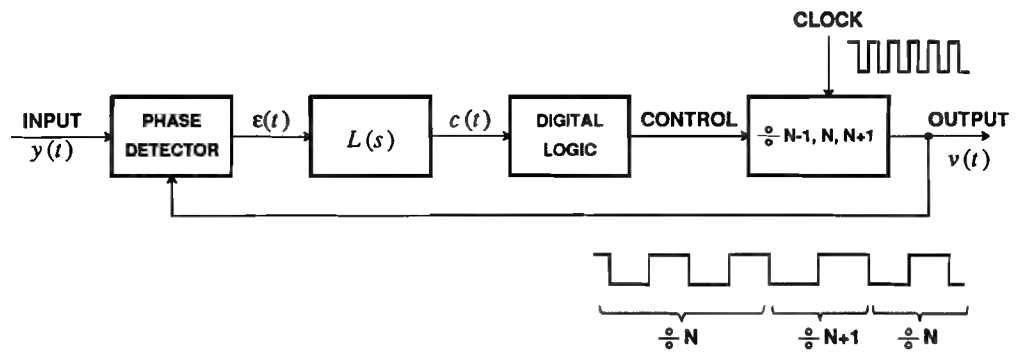
## 15.4. VARIATIONS ON A THEME: VCOs

Just as with phase detectors, there are many variations on the design of VCOs. We describe two important examples.

### 15.4.1. Digital VCOs

Analog VCOs are complicated circuits, difficult to design and sensitive to environmental factors such as temperature. A frequently used alternative is the digital VCO, shown in Figure 15-15. The VCO produces a square wave that is generated from a local high frequency clock using a variable countdown chain (a frequency divider). By controlling the frequency divider, the frequency of the VCO output can be controlled. The digital logic is such that if  $c(t) < -K$ , for some threshold  $K$ , the frequency of  $v(t)$  is decreased by dividing by  $N + 1$  instead of the nominal  $N$ . Similarly, if  $c(t) > K$ , the divider divides by  $N - 1$  to increase the output frequency. In a typical operating condition,  $c(t)$  is hovering near  $\pm K$ , and the frequency divider is either alternately dividing by  $N$  and  $N + 1$  or  $N - 1$  and  $N$ .

For digital VCOs, the lock range is not determined by the  $2\pi$  ambiguity of the phase detector, but rather by  $N$ , the nominal divide ratio of the VCO. The reason is obvious: the maximum frequency the VCO can produce occurs when the divider is always dividing by  $N - 1$ , and the minimum frequency occurs when the VCO is dividing by  $N + 1$ . Thus any input frequency outside this range cannot be held (see Problem 15-15).



**Figure 15-15.** A digital VCO uses a controllable countdown chain (a frequency divider) to generate the output, a square wave.

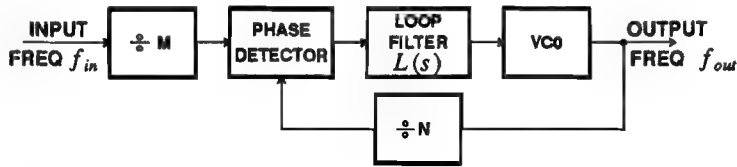
Although digital VCOs are economical and are commonly used, for some applications they have a serious disadvantage. Namely, since the frequency divider is alternating between two divide ratios at all times, the square wave at its output has considerable jitter. This jitter can be minimized for large  $N$ , but large  $N$  implies a high frequency oscillator. Depending on the nominal frequency  $\omega_v$  of the VCO, it may not be practical to implement an oscillator and frequency divider at frequency  $N\omega_v$  for  $N$  large enough to produce acceptably low jitter. In these situations, it is necessary to either develop algorithms that are insensitive to jitter, or resort to the use of an analog PLL. For a practical example of such a situation, see [10,11,12].

**15.4.2. Frequency Synthesizers**

It is possible to design PLLs that maintain phase lock when the VCO output frequency is not equal to the input frequency, but is related by a fixed rational multiple. Such a PLL is called a *frequency synthesizer*, shown in Figure 15-16. When the PLL is phase locked,

$$\frac{f_{in}}{M} = \frac{f_{out}}{N} , \quad f_{out} = \frac{N}{M} f_{in} . \tag{15.69}$$

Frequency synthesizers are widely used in multiplexers, where it is often necessary to



**Figure 15-16.** A frequency synthesizer produces an output signal with frequency equal to  $N/M$  times the input frequency.

synchronously generate one frequency from another.

### Example 15-15.

Given a clock at 1,544 kHz, what are  $M$  and  $N$  such that we generate a clock at 2,048 kHz? We could use  $N = 2,048$  and  $M = 1,544$ , and the inputs to the phase detector would be on the order of 1 kHz. Suppose that an exclusive-or phase detector (Figure 15-13) is used. Then the output  $x(t)$  of the exclusive-or will be periodic with frequency 1 kHz (see Figure 15-13b). The LPF in Figure 15-13a has to have sufficiently narrow bandwidth to remove this fundamental and its harmonics. In addition to complicating the design of the LPF, the narrow bandwidth means that the PLL will respond slowly to changing conditions.

This design is easily improved so that the bandwidth of the LPF can be larger. Observe that  $1,544 = 193 \times 8$  and  $2,048 = 256 \times 8$ . Consequently we can use  $N = 256$  and  $M = 193$ . This results in inputs to the phase detector on the order of 8 kHz, significantly relaxing the specifications of the LPF.  $\square$

## 15.5. FURTHER READING

The understanding of noise performance, acquisition, and non-linear behavior of PLLs is advanced, and we have largely omitted these topics. The available literature, however, is heavily biased towards continuous-time, analog PLLs. In digital communications, many PLLs are either entirely discrete-time or a mixture of discrete and continuous-time. Fortunately, extending the continuous-time results to discrete time is often easy, as illustrated in Section 15.2.

A number of books give comprehensive coverage of analog, continuous-time PLLs. Gardner [2] was the first widely used for design purposes. The first four chapters of Viterbi [13] are devoted to PLL theory. The third chapter gives a particularly good description of an analysis technique called phase-plane analysis. Van Trees [14] uses non-linear estimation theory for the analysis of PLLs. Klapper and Frankle [15] give a detailed treatment biased towards the application of PLLs as FM discriminators. Lindsey [16] gives a thorough theoretical treatment with emphasis on weak signal applications. A more recent text by Blanchard [17], emphasizes design of coherent receivers for analog modulation schemes. Two other important books are by Best [18] and Lindsey and Simon (editors) [19]. A brief overview of advanced PLL topics is given by Gupta [20] with an extensive bibliography. The October 1982 issue of IEEE Transactions on Communications is devoted to PLLs, providing an excellent source of more up-to-date papers. A description of frequency detectors is given by Messerschmitt [1]. Of historical importance is perhaps the first paper on PLLs by Appleton in 1922 [21]. A relatively recent innovation is the adaptive PLL, in which the loop filter is adaptive rather than fixed [22,23]. The filter can adjust itself to changing phase characteristics on the input.

## PROBLEMS

15-1. Consider the PLL shown in Figure 15-2. Suppose  $L(s) = K_L/s$ .

- (a) What is the order of the loop?
- (b) Is the loop stable? Why?

15-2. The lock range of the first-order PLL in Example 15-4 can be increased by increasing the loop gain  $K_L$ . Show that as  $K_L$  tends to infinity the output of the VCO will achieve perfect phase lock for any input phase  $\theta(t)$ . What happens to the bandwidth of the PLL? When might this be useful, and when might it not be useful?

15-3.

- (a) Show that  $s^2 + as + b$  has all roots in the open left half plane if and only if the real-valued coefficients satisfy  $a > 0$  and  $b > 0$ . This basic result can be used to determine the stability of any second-order continuous-time PLL.
- (b) Show that an ideal PLL with loop filter given by (15.19) is stable if and only if  $K_2 > -K_L$  and  $K_L K_1 > 0$ .

15-4. Suppose the loop filter of a continuous-time second-order PLL with an ideal VCO and phase detector is given by

$$L(s) = \frac{1}{s + \sqrt{2}}. \quad (15.70)$$

- (a) Find the poles of the phase transfer function. Is the PLL stable?
- (b) Show that the gain  $|\Phi(j\omega)/\Theta(j\omega)|$  is never greater than unity, so there is no peaking.
- (c) Sketch the magnitude of the frequency response.

15-5. Consider a type I second-order PLL with the loop filter given by (15.19) and

$$K_L = K_2 = 1 \quad K_1 = 0.5. \quad (15.71)$$

- (a) Give the closed loop transfer function to phase  $\Phi(s)/\Theta(s)$ .
- (b) Is the PLL stable?
- (c) Sketch the Bode plot for phase transfer function.
- (d) Show that the PLL does not exhibit peaking.

15-6. For the type I second-order PLL of (15.19),

- (a) Show that there is peaking if and only if

$$K_2^2 \leq 2K_L K_1. \quad (15.72)$$

- (b) Find the range of frequencies  $\omega$  for which the gain is greater than unity,  $|\Phi(j\omega)/\Theta(j\omega)|^2 > 1$ .
- (c) The bandwidth of a type I second-order PLL with complex poles can be estimated as  $\sqrt{K_{L2} K_1}$  (see Figure 15-5) where  $K_{L2}$  is  $K_L$  for the second-order PLL. By contrast, the bandwidth of a first-order PLL can be estimated as  $K_{L1}$ , which is  $K_L$  for the first-order loop. Assume these two PLLs have equal bandwidth,

$$\sqrt{K_{L2} K_1} = K_{L1}. \quad (15.73)$$

Show that if there is no peaking, then the lock range of the second-order loop is actually *smaller* than the lock range of the first-order loop.

15-7.

- (a) Use the result stated in Problem 15-6a to show that a stable type II second-order PLL *always* exhibits peaking.

- (b) Show that

$$\left| \frac{\Phi(j\omega_n)}{\Theta(j\omega_n)} \right|^2 = \frac{K_1 + K_L}{K_L}, \quad (15.74)$$

where  $\omega_n = \sqrt{K_L K_1}$ , the natural frequency. We see that the magnitude of the peaking at  $\omega_n$  can be made small by choosing a small  $K_1$ .

- (c) Show that this type II loop will be stable if and only if  $K_1$  and  $K_L$  are each greater than zero. This observation combined with (15.74) is another way of showing that this PLL always exhibits peaking, but it also shows that the gain at  $\omega_n$  can be made as close to unity as we like by selecting a small  $K_1$ .

- 15-8. Consider the type I second-order PLL with a loop filter of the form

$$L(s) = \frac{K_1}{K_2 + s}. \quad (15.75)$$

This is different from (15.19) in that there is no zero for finite  $s$ .

- (a) Find the transfer function and show that it is a lowpass filter with unity gain at d.c.  
 (b) Show that this PLL is stable if and only if  $K_1 > 0$  and  $K_2 > 0$ .  
 (c) Find the range of frequencies  $\omega$  over which components  $\Theta(j\omega)$  in the input phase will be amplified. Under what conditions is there no amplification (peaking)?
- 15-9. Consider the PLL in Figure 15-2. Suppose that  $L(s) = K_L$ , a positive constant, and  $\theta(t)$  is given below. Find the steady-state response

$$\epsilon_{ss} = \lim_{t \rightarrow \infty} \epsilon(t). \quad (15.76)$$

- (a)  $\theta(t) = \beta u(t)$ , where  $u(t)$  is the unit step function.  
 (b)  $\theta(t) = \beta t^2 u(t)$ .

Compare these results to the result derived in Example 15-9 for  $\theta(t) = \omega_0 t u(t)$ .

- 15-10. Consider the PLL in Figure 15-2. Show that all loop filters  $L(s)$  with one or more poles at  $s = 0$  satisfy  $\epsilon_{ss} = 0$  when  $\theta(t) = \omega_0 t u(t)$ , which corresponds to a steady frequency offset of  $\omega_0$ . Assume  $L(s)$  is chosen so that the PLL is stable.
- 15-11. Consider the first-order discrete-time PLL of Example 15-10. Sketch the magnitude of the frequency response  $|\Phi(e^{j\omega T})/\Theta(e^{j\omega T})|$  when:
- (a)  $K_L = 0.5$   
 (b)  $K_L = 1$   
 (c)  $K_L = 1.5$ .  
 (d) Compare the relative merits of the PLLs in (a) through (c).
- 15-12. Consider the second-order polynomial with real coefficients

$$D(z) = z^2 + az + b = (z - p)(z - q). \quad (15.77)$$

Show that the roots  $p$  and  $q$  lie inside the unit circle (have less than unity magnitude) if and only if  $|b| = |pq| < 1$  and  $|a| < 1 + b$ .

- 15-13. Use the result proven in Problem 15-12 to find conditions under which the discrete-time PLL with loop filter given by (15.46) is stable.
- 15-14. Consider a continuous-time PLL with a sinusoidal phase detector, Figure 15-9. Suppose that this PLL is in perfect phase lock,  $\theta(t) = \phi(t)$  (not quadrature phase lock). Find an expression for the input phase  $\theta(t)$  as a function of the PLL parameters and time.

- 15-15. Consider the digital VCO in Figure 15-15. Suppose that the clock frequency is 1 MHz and  $N = 100$ . Suppose that the input has the form  $\cos(\omega_1 t + \theta)$  where  $\theta$  is a constant. Find the range of  $\omega_1$  such that the PLL can maintain phase lock.
- 15-16. Consider the design of a frequency synthesizer (Figure 15-16) that synthesizes a 2,048 kHz signal given a 1,512 kHz input.
- Find the minimum values of  $M$  and  $N$ .
  - Consider doing the frequency synthesis with two cascaded frequency synthesizers. Select  $M_1$ ,  $N_1$ ,  $M_2$  and  $N_2$  for each of the frequency dividers. What advantages does this design have over the one with a single frequency synthesizer?
  - What is the maximum number of cascaded frequency synthesizers that can be used effectively?

## REFERENCES

- D. G. Messerschmitt, "Frequency Detectors for PLL Acquisition in Timing and Carrier Recovery," *IEEE Trans. on Communications* COM-27(9) p. 1288 (Sep. 1979).
- F. M. Gardner, *Phaselock Techniques*, Wiley, New York (1966).
- G. S. Gill and S. C. Gupta, "First-Order Discrete Phase-Locked Loop with Applications to Demodulation of Angle-Modulated Carrier," *IEEE Trans. Communication Technology* COM-20 pp. 454-462 (June 1972).
- G. S. Gill and S. C. Gupta, "On Higher Order Discrete Phase-Locked Loops," *IEEE Trans. Aerospace Electronic Systems* AES-8 pp. 615-623 (Sep. 1972).
- A. Weinberg and B. Liu, "Discrete Time Analyses of Non-Uniform Sampling First- and Second-Order Digital Phase-Locked Loops," *IEEE Trans. Communications Technology* COM-22 pp. 123-137 (Feb. 1974).
- F. M. Gardner, "Properties of Frequency Difference Detectors," *IEEE Trans. Communications* COM-33 pp. 131-138 (Feb. 1985).
- J. C. Haartsen and R. C. Den Dulk, "Novel Circuit Design and Implementation of Adaptive Phase Comparators," *Electronics Letters* 23(11) pp. 551-552 (May 1987).
- J. Eijssendoorn and R. C. Den Dulk, "Improved Phase-Locked Loop Performance with Adaptive Phase Comparators," *IEEE Trans. Aerospace and Elec. Sys.* AES-18 pp. 323-333 (May 1982).
- J. F. Oberst, "Generalized Phase Comparators for Improved Phase-Lock Loop Acquisition," *IEEE Trans. Communications* COM-19 pp. 1142-1148 (Dec. 1971).
- D. D. Falconer, "Timing Jitter Effects on Digital Subscriber Loop Echo Cancellers: Part I - Analysis of the Effect," *IEEE Trans. on Communications* COM-33(8)(Aug. 1985).
- D. D. Falconer, "Timing Jitter Effects on Digital Subscriber Loop Echo Cancellers: Part II - Considerations for Squaring Loop Timing Recovery," *IEEE Trans. on Communications* COM-33(8)(Aug. 1985).
- D. G. Messerschmitt, "Asynchronous and Timing-Jitter Insensitive Data Echo Cancellation," *IEEE Trans. on Communications* COM-34(12) p. 1209 (Dec. 1986).
- A. J. Viterbi, *Principles of Coherent Communication*, McGraw-Hill, New York (1966).
- H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part II*, Wiley, New York (1971).

15. J. Klapper and J. T. Frankle, *Phase-Locked and Frequency-Feedback Systems*, Academic Press, New York (1972).
16. W. C. Lindsey, *Synchronization Systems in Communications*, Prentice-Hall, Englewood Cliffs, N.J. (1972).
17. A. Blanchard, *Phase-Locked Loops: Application to Coherent Receiver Design*, John Wiley and Sons, New York (1976).
18. R. E. Best, *Phase-Locked Loops; Theory, Design, and Applications*, McGraw Hill, New York (1984).
19. W. C. Lindsey and M. K. Simon, *Phase-Locked Loops and Their Applications*, IEEE Press, New York (1978).
20. S. C. Gupta, "Phase-Locked Loops," *Proceedings of the IEEE* 63(2)(Feb. 1975).
21. E. V. Appleton, "Automatic Synchronization of Triode Oscillators," *Proc. Cambridge Phil. Soc.* 21 p. 231 (1922-1923).
22. R. P. Gooch and M. J. Ready, "An Adaptive Phase Lock Loop for Phase Jitter Tracking," *Proceedings of the 21st Asilomar Conf. on Signals, Systems, and Computers*, (Nov. 2-4, 1987).
23. R. D. Gitlin, "Adaptive Phase-Jitter Tracker," *United States Patent, Patent No. 4,320,526*, (March 16, 1982).

# 16

---

## CARRIER RECOVERY

---

In passband systems, the carrier frequency is generated in the transmitter from a local timing reference such as a crystal oscillator. As we saw in Chapters 6 and 8, *coherent demodulation* of a passband signal requires exactly the same carrier frequency and phase to perform the demodulation. But the receiver usually has an independent timing reference, as illustrated in Figure 15-1. Deriving the carrier frequency and phase from the data bearing signal is the topic of this chapter.

In previous chapters we have assumed that both the symbol timing and the carrier frequency are known at the receiver. In this chapter we will assume the symbol timing is known, and derive the carrier frequency. Chapter 15 will show that symbol timing can be derived without knowledge of the carrier phase. Hence, when a receiver first starts receiving data, it should first derive timing using the techniques of Chapter 17, then estimate the carrier phase using the techniques in this chapter, and finally adapt the equalizer (Chapter 11).

If the symbol timing is known, it may be that the carrier frequency can be derived from it. If the carrier frequency used at the transmitter is a fixed rational multiple of the symbol rate, then the frequency synthesizer of Figure 15-16 can derive a high quality carrier, even if there is considerable jitter on the derived timing signal. However, even if the transmitter uses a carrier frequency and symbol rate that are related by a rational multiple, that relationship may be lost by the time they get to the receiver.



**Example 16-1.**

As described in Section 5.5, telephone channels often introduce frequency offset. It is shown in Exercise 5-6 that if the passband PAM signal

$$x(t) = \text{Re}\{s(t)e^{j\omega_c t}\} \quad (16.1)$$

is subjected to frequency offset of  $\omega_0$  (and no other impairments) then the received signal will be

$$y(t) = \text{Re}\{s(t)e^{j(\omega_c - \omega_0)t}\}. \quad (16.2)$$

Frequency offset therefore is indistinguishable from using a different carrier frequency  $\omega_c - \omega_0$ . The symbol rate, however, cannot be changed by the channel because the receiver can only receive exactly as many symbols per unit time as are sent. Consequently, even if the symbol rate and carrier frequency start out as rational multiples of one another at the transmitter, their relationship at the receiver is dependent on the unknown frequency offset.

□

**Example 16-2.**

In microwave radio applications where either the transmitter or the receiver is in motion, the carrier frequency is subject to a Doppler shift, but the symbol timing is not. The resulting frequency offset is similar to that found in telephone channels, although it is more likely to be time-varying as the velocity changes. □

In addition to frequency offset, it is common for a channel to introduce *phase jitter* which appears as fluctuations in the phase of the carrier. It is desirable to track the phase jitter so that it does not degrade the performance of the system. So even in the absence of frequency offset, it is still desirable to derive the carrier independently from the timing so that phase jitter can be tracked.

We will describe two techniques for tracking the carrier in the receiver. The first is a decision-directed technique, and the second is the *power of N* method.

## 16.1. DECISION-DIRECTED CARRIER RECOVERY

Consider a noiseless passband PAM analytic signal that has been subjected to frequency offset and/or phase jitter

$$e^{j(\omega_c t + \theta(t))} \sum_{m=-\infty}^{\infty} A_m p(t - mT) \quad (16.3)$$

where  $p(t)$  accounts for the transmit pulse shape, the linear distortion in the channel, and the receive filter, and  $\theta(t)$  models the frequency offset and phase jitter. In order to have this analytic signal available at the receiver, we assume either an analytic receiver bandpass filter or a phase splitter, as discussed in Section 6.4. If there is frequency offset then  $\theta(t)$  will have a linear term  $\omega_0 t$ . Suppose that (16.3) is demodulated with the carrier

$$e^{-j(\omega_c t + \phi(t))} \quad (16.4)$$

where  $\phi(t)$  is the receiver estimate of the carrier phase. Sample at the symbol rate  $t = kT$  to get

$$q_k = e^{j(\theta_k - \phi_k)} \sum_{m=-\infty}^{\infty} A_m p_{k-m}, \quad (16.5)$$

where  $\theta_k$ ,  $\phi_k$ , and  $p_k$  are samples of  $\theta(t)$ ,  $\phi(t)$ , and  $p(t)$ . If the carrier recovery follows a bandpass equalizer, as shown in Figure 6-23, then the equalized pulse shape should approximately satisfy the Nyquist criterion,

$$p_k = \delta_k, \quad (16.6)$$

and consequently

$$q_k = e^{j(\theta_k - \phi_k)} A_k. \quad (16.7)$$

### Example 16-3.

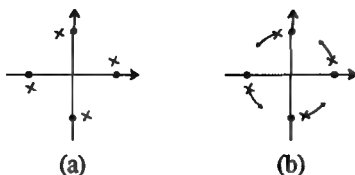
If the receiver demodulates with a constant phase error  $\theta_k - \phi_k = \psi$ , and there is no other degradation, then the received constellation will be a tilted version of the transmitted constellation, as shown in Figure 16-1a.  $\square$

If left uncorrected, the tilt will degrade the immunity of the receiver to noise by bringing the received signal points closer to the boundaries of the decision regions.

### Example 16-4.

If the receiver demodulates with the wrong frequency  $\theta_k - \phi_k = \omega_0 kT$ , then the received constellation rotates, as illustrated in Figure 16-1b.  $\square$

If left uncorrected, the rotating constellation will make errors every time a received symbol rotates past the boundary of a decision region. To correct both problems, carrier recovery is needed. Assuming a PLL is used for this function, we will now design a decision-directed phase detector. It is also possible to avoid decision direction in carrier recovery, as illustrated in Section 16.2.



**Figure 16-1.** If the receiver demodulates with a constant phase error then the received constellation will be a tilted version of the transmitted constellation, as shown in (a). The O's are the transmit constellation and the X's are the received symbols. If the receiver demodulates with the wrong frequency, the received constellation will rotate, as illustrated in (b).

**Exercise 16-1.**

Given (16.7), show that the phase error in the demodulator is

$$\epsilon_k = \theta_k - \phi_k = \sin^{-1} \left[ \frac{\text{Im} \{ q_k A_k^* \}}{|A_k|^2} \right]. \quad (16.8)$$

□

We have the beginnings of a phase detector, but in practical systems the assumptions that  $p_k = \delta_k$  and that there is no noise are unrealistic. With any amount of noise and intersymbol interference we can write the received symbols as

$$q_k = c_k e^{j\epsilon_k} A_k \quad (16.9)$$

where the real-valued  $c_k > 0$  accounts for amplitude errors and  $\epsilon_k$  accounts for phase errors. Some of the phase error will be due to noise and intersymbol interference, and some will be due to phase jitter and frequency offset.

**Exercise 16-2.**

Given (16.9), show that

$$\epsilon_k = \sin^{-1} \left[ \frac{\text{Im} \{ q_k A_k^* \}}{|q_k| \cdot |A_k|} \right]. \quad (16.10)$$

□

We are now closer to a practical phase detector, but there is still a problem. The symbols  $A_k$  are not known in the receiver (or there would be no need to transmit them) except perhaps during a brief training period at the initiation of a connection. Just as we did with adaptive equalizers in Chapter 11, we can use decisions  $\hat{A}_k$  instead of the actual symbols  $A_k$ . The resulting carrier recovery loop is shown in Figure 16-2.

This PLL is closely related to those in Chapter 15, of course, but the phase detector is significantly different. One major difference: since the phase detector is decision-directed, errors in the decision will result in errors in phase detection.

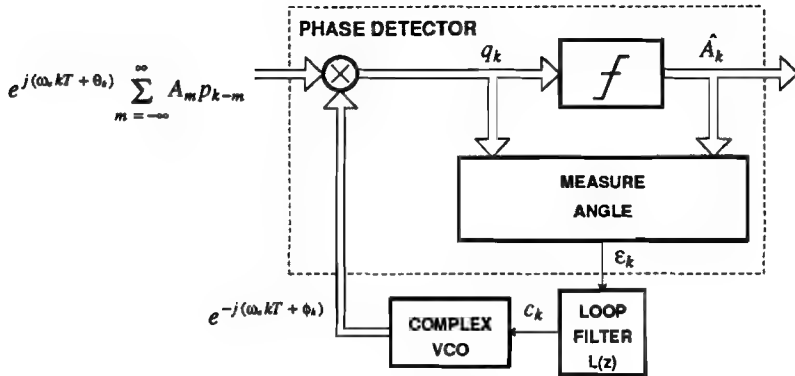
**Example 16-5.**

Consider a 4-PSK signal with constellation and decision regions shown in Figure 16-3a. If the received sample has a phase error greater than  $\pi/4$  in magnitude, the decision will be incorrect, and the measured phase error will be incorrect. Equivalently, the measured phase error is

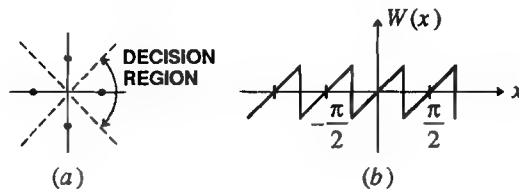
$$\epsilon_k = W(\theta_k - \phi_k) \quad (16.11)$$

where  $W(\cdot)$  is shown in Figure 16-3b. An immediate consequence is that the derived carrier can have any of four phases, depending on the first few decisions. □

Decision-directed carrier recovery generally suffers from a phase ambiguity in the derived carrier, as shown in the previous example. This problem is easily overcome by using a *differential encoder*, much like those used for AMI in Section 12.1.



**Figure 16-2.** A carrier recovery loop using a phase detector that measures the angular difference between the received sample  $q_k$  and the decision  $\hat{A}_k$ .



**Figure 16-3.** a. The constellation and decision regions for a 4-PSK signal. b. The characteristic of an ideal decision-directed phase detector. It cannot detect angular errors greater than  $\pi/4$  in magnitude.

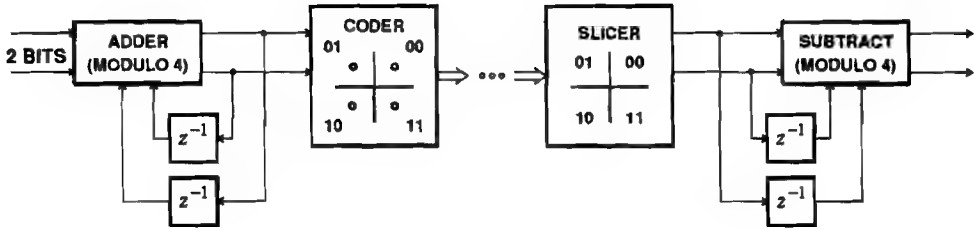
### Example 16-6.

A differential encoder and decoder for a 4-PSK signal is shown in Figure 16-4. Information is carried by the change in phase, rather than by the absolute phase. For example, an increase in the transmitted phase by  $\pi$  indicates the transmission of the pair of bits 10, regardless of the absolute phase.  $\square$

### Exercise 16-3.

Assume the only degradation between the coder and the slicer in Figure 16-4 is a rotation by a multiple  $M$  of  $\pi/2$ , or in other words a multiplication by  $e^{jM\pi/2}$ . Such a degradation could be introduced by carrier recovery that uses a phase detector with the characteristic in Figure 16-3b. Show that the output bits from the decoder are the same regardless of the multiple  $M$  (with the possible exception of the first two bits out of the decoder).  $\square$

Proper operation of a differential decoder depends on the decisions being correct in the receiver. In fact, if a single decision is incorrect, two symbol intervals will be incorrectly decoded, so there is *error propagation*. The impact of this error propagation is usually minimal.



**Figure 16-4.** A differential encoder can be used in the transmitter to overcome the  $\pi/2$  phase ambiguity in the carrier recovery loop. Information is carried by the change in phase, rather than by the absolute phase. The decoder shown reverses the encoding.

For constellations larger than 4-PSK, the phase ambiguity of the carrier recovery loop can be more serious (see Problem 16-2). It is common to differentially encode two bits of the  $M$  bits needed for a constellation of size  $2^M$ . The coder then uses the two differentially encoded bits to determine the quadrant. Thus phase errors of multiples of  $\pi/2$  do not cause decision errors, although smaller phase errors might.

The phase error measurement in (16.10) can be simplified. Observe that for small phase errors,  $\epsilon_k \approx \sin(\epsilon_k)$ , and

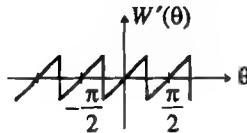
$$\sin(\epsilon_k) = \frac{\text{Im}\{q_k A_k^*\}}{|q_k| \cdot |A_k|}. \quad (16.12)$$

We can write this as

$$\sin(\epsilon_k) = W'(\theta_k - \phi_k) = \sin(W(\theta_k - \phi_k)) \quad (16.13)$$

where  $W'(\cdot)$  is shown in Figure 16-5 and  $W(\cdot)$  is shown in Figure 16-3b for the 4-PSK case. From Figure 16-5 we see that (16.12) makes a reasonable phase detector. With small angular errors, the characteristic is close to linear. It is common to simplify still further and omit the denominator in (16.12) so as to avoid having to perform the division, using as the estimate of the phase error

$$\epsilon'_k = \text{Im}\{q_k A_k^*\}. \quad (16.14)$$

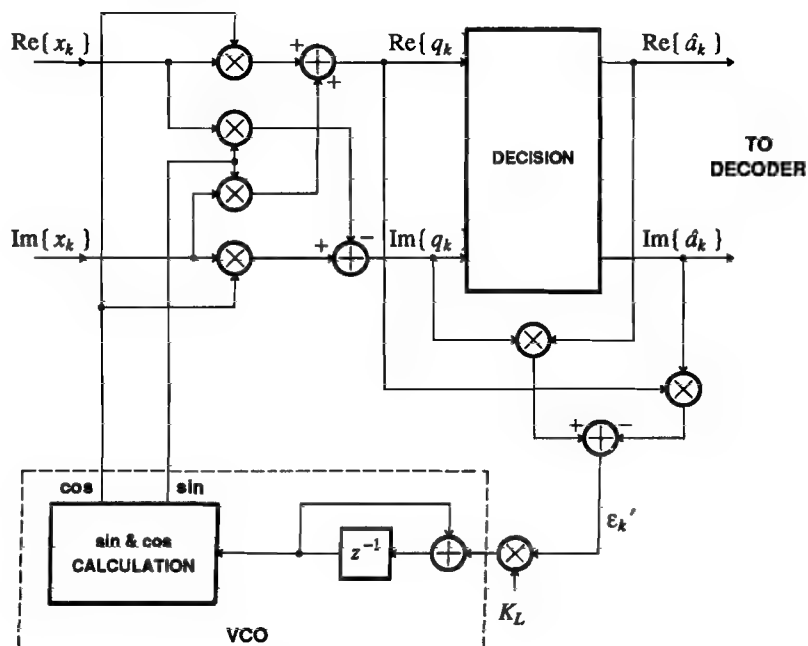


**Figure 16-5.** The phase detector characteristic when (16.12) is used to estimate the angular error. Except for a slight curvature, it is very similar to that in Figure 16-3b.

A first-order carrier recovery loop ( $L(z) = K_L$ ) using this angular error estimate is shown in Figure 16-6. The complex multiplications are shown explicitly so that the total complexity of the algorithm can be understood at a glance. This carrier recovery loop, or a variant of it with a continuous-time VCO, is commonly used, and is quite effective.

Many of the techniques from Chapter 15 can be applied to adapt this basic technique to particular situations. For example, a natural extension of the basic PLL in Figure 16-6 uses a higher order loop, obtained by inserting a more complicated filter  $L(z)$ . Careful design of  $L(z)$  can lead to carrier recovery loops that perfectly track frequency offset or phase jitter at some frequency.

One practical difficulty arises when an adaptive equalizer is used in conjunction with a decision-directed carrier recovery loop. Baseband adaptive equalizers assume that the input has been demodulated. The solution to this difficulty, given in Section 11.5, is to use a passband equalizer. We can now understand why this is so important. The overall structure of a passband PAM receiver is shown in Figure 6-23. By placing the forward equalizer before the carrier recovery demodulation, we avoid having the equalizer inside the carrier recovery loop. By contrast, a baseband equalizer would follow the demodulator and precede the slicer. This means that it is inside the carrier recovery loop. Consequently, the loop transfer function of the carrier recovery



**Figure 16-6.** A first-order carrier recovery loop using (16.14), a particularly simple estimate of the angular error.

includes the time-varying equalizer, causing considerable complication. At the very least, the long delay (several symbol intervals) associated with the baseband equalizer would force the loop gain of the carrier recovery to be reduced to ensure stability, impairing its ability to track rapidly varying carrier phase. The passband equalizer shown in Figure 6-23 mitigates this problem by equalizing prior to demodulation.

Another important practical difficulty arises with decision-directed carrier recovery loops when trellis coding is used (Chapter 14). Unlike a slicer, a trellis decoder does not make immediate decisions. Decisions may be postponed several (say  $M$ ) symbol intervals. This is equivalent to inserting a delay  $z^{-M}$  in the carrier recovery loop. The delay can undermine the validity of the PLL (see Problem 16-3). To overcome this practical difficulty, a slicer is added to the receiver and the slicer decisions are used to compute the phase error. The slicer decisions will not be as accurate as the decisions made by the trellis decoder, but at least they are made promptly.

### Fixed Reference Detector

A particularly simple variation on decision-directed carrier recovery is possible for PSK signals. It is a phase-domain PLL like that in Figure 15-14, but modified as shown in Figure 16-7. The phase of the input signal is compared against a fixed reference generated by a local oscillator. This phase difference (which is modulo  $2\pi$ ) will reflect the phase modulation of the input plus a drift with time due to the fact that the local oscillator is not synchronized with the remote carrier. The function of the PLL is to remove the drift at the first subtractor. The slicer determines which of  $N$  phases was transmitted using a simple threshold test, and the difference between the decision and the input to the slicer is a measurement of the residual drift  $\epsilon_k$  (or residual phase error). This phase error is filtered (if desired), and integrated. The integrator comes from the structure in Figure 15-14. The main advantage of the fixed reference detector in Figure 16-7 is that it can be implemented using inexpensive and fast digital logic.

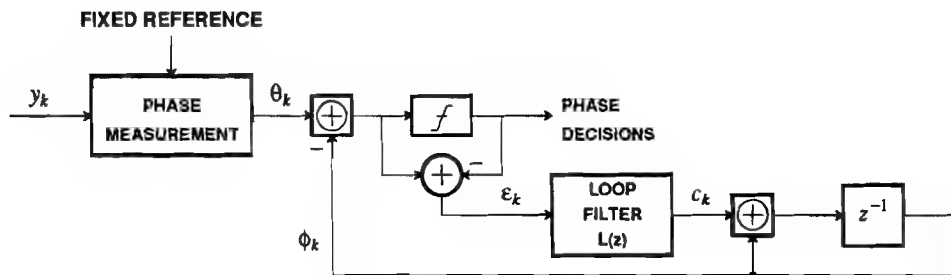


Figure 16-7. A fixed-reference detector for PSK signals.

## 16.2. POWER OF N CARRIER RECOVERY

Although decision-directed carrier recovery is a common technique for coherent demodulation, decision errors will be a problem at low SNR. In this event it is possible to avoid decision direction. We will illustrate this with a popular alternative, *power of N carrier recovery*. Consider again the sampled passband PAM analytic signal

$$\begin{aligned} x_k &= e^{j(\omega_c kT + \theta_k)} \sum_{m=-\infty}^{\infty} A_m p_{k-m} \\ &= e^{j(\omega_c kT + \theta_k)} \sum_{m=-\infty}^{\infty} |A_m| e^{j \arg(A_m)} p_{k-m}. \end{aligned} \quad (16.15)$$

Assume that there is no ISI so that  $p_k = \delta_k$  and

$$x_k = e^{j(\omega_c kT + \theta_k)} |A_k| e^{j \arg(A_k)}. \quad (16.16)$$

If we raise this signal to the  $N^{\text{th}}$  power we get

$$x_k^N = e^{jN(\omega_c kT + \theta_k)} |A_k|^N e^{jN \arg(A_k)}. \quad (16.17)$$

Now suppose that we can find an integer  $N$  such that

$$e^{jN \arg(A_k)} = 1 \quad (16.18)$$

for all  $A_k$ . For example, this is possible for PSK and AM-PM signals. Then

$$x_k^N = e^{jN(\omega_c kT + \theta_k)} |A_k|^N. \quad (16.19)$$

This signal has a strong spectral line at frequency  $N\omega_c$ , as is evident from the following breakdown,

$$x_k^N = e^{jN(\omega_c kT + \theta_k)} \mathbb{E}[|A_k|^N] + e^{jN(\omega_c kT + \theta_k)} \left[ |A_k|^N - \mathbb{E}[|A_k|^N] \right]. \quad (16.20)$$

The first term is a tone that can be extracted with a bandpass filter or a PLL. The second term may be zero (e.g. for PSK), or may contribute amplitude modulation to the tone.

Except for the possibly time-varying amplitude term  $|A_k|^N$ , (16.19) has the same form that we assumed in Chapter 15 for the complex phase detector (see Figure 15-11). It can be fed into a complex phase detector, as shown in Figure 16-8. The natural frequency  $\omega_v$  of the VCO should be selected so that  $N\omega_c$  is always within the lock range of the PLL (see Problem 16-5).

### Example 16-7.

Consider a 4-PSK signal with  $|A_k| = 1$ , and no ISI or noise, just phase jitter and frequency offset represented by  $\theta_k$  in (16.15). Let  $N = 4$  so from (16.19)

$$x_k^N = e^{j4(\omega_c kT + \theta_k)}. \quad (16.21)$$

In the presence of noise or ISI this signal will have phase jitter which can be attenuated by the PLL.  $\square$



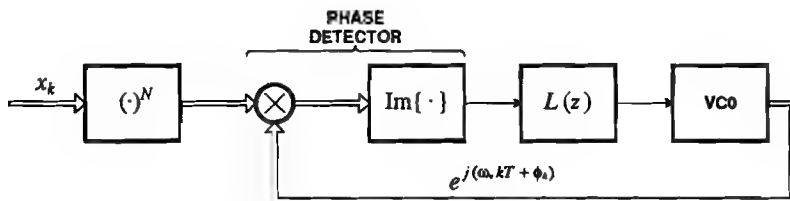


Figure 16-8. A power of  $N$  carrier recovery loop using the complex phase detector of Figure 15-11.

To demodulate the PAM signal, the complex sinusoid with frequency  $N\omega_c$  must be converted to a complex sinusoid at frequency  $\omega_c$ . This can be done using the frequency synthesizer in Figure 15-16, which can actually be incorporated into the loop in Figure 16-8 (see Problem 16-5).

In practical situations, of course, noise and ISI will be present and will contribute to phase jitter in the derived carrier. This jitter can be attenuated with a narrowband filter or a PLL.

Another practical difficulty is that it may not always be possible to find a value of  $N$  such that (16.18) is true (see Problem 16-4). In this case, we can raise the signal to the  $N^{\text{th}}$  power anyway as in (16.17) and the breakdown similar to (16.20) becomes

$$x_k^N = e^{jN(\omega_c kT + \theta_k)} E[A_k^N] + e^{jN(\omega_c kT + \theta_k)} \left[ A_k^N - E[A_k^N] \right]. \quad (16.22)$$

The first term will yield the desired tone as long as  $E[A_k^N] \neq 0$ . It is usually easy to find an  $N$  for which this is true (see Problem 16-4).

Not surprisingly, continuous-time versions of power of  $N$  timing recovery are also used (see Problem 16-6).

### 16.3. FURTHER READING

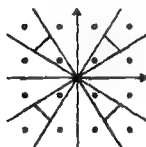
A useful tutorial on carrier and timing recovery is given by Franks [1]. The interaction of carrier recovery and adaptive equalization was resolved in the seminal paper by Falconer [2]. Further information about decision-directed carrier recovery can be obtained from [3,4,5,6,7]. An interesting variation of 4-th power carrier recovery is given in [8]. Simultaneous maximum likelihood carrier and timing recovery is considered in [9]. The use of an adaptive loop filter for carrier recovery is proposed in [10,11].

# PROBLEMS

- 16-1.** A common problem that must be avoided in carrier recovery is *false lock*. For example, in Figure 16-1b, for four-phase PSK if the constellation rotates precisely  $\pi/2$  radians in every sample time, the samples  $q_k$  will fall on the correct signal constellation points and it will be difficult to tell that the carrier frequency is off.
- Find the carrier offset frequencies that result in this false lock.
  - For a carrier frequency of 75 MHz and a baud rate of 15 MHz, what is the minimum false-lock carrier-offset frequency?
  - Suggest one or more possible ways to deal with this problem.
- 16-2.** Suppose that a decision-directed phase detector given by (16.10) (with  $A_k$  replaced by the decisions  $\hat{A}_k$ ) is used in a carrier recovery loop. We can write  $\epsilon_k$  as a function of the phase error,

$$\epsilon_k = W(\theta_k - \phi_k). \quad (16.23)$$

- Sketch  $W(\cdot)$  for an 8-PSK constellation.
- Sketch  $W(\cdot)$  for a 16-QAM constellation. Assume radial decision regions, as shown in the following figure:



The decision boundary angles are halfway between the angles of the symbols. (These decision regions are far from optimal, but they simplify the problem.) **Hint:**  $W(\cdot)$  depends on  $A_k$ .

- 16-3.** Consider the discrete-time carrier recovery loop in Figure 16-2 with a first-order loop filter  $L(z) = K_L$ . Suppose that the decisions made by the slicer are delayed by  $M$  samples (for example the slicer is replaced with a trellis decoder that has truncation depth  $M$ ). This extra delay can be modeled as a loop filter

$$L(z) = z^{-M} K_L \quad (16.24)$$

instead of  $K_L$ . Suppose that  $K_L = 0.1$ .

- For no delay,  $M = 0$ , find the pole location of the transfer function to jitter  $\Phi(z)/\Theta(z)$  and sketch the frequency response.
  - Repeat for  $M = 1$ . Is this PLL useful?
  - Find the poles when  $M = 2$ .
  - Find  $M$  such that the loop becomes unstable (may require a computer program).
- 16-4.**
- Show that there is no value of  $N$  such that (16.18) is satisfied for a 16-QAM signal.
  - Find  $N$  such that  $E[|A_k|^N] \neq 0$  for a 16-QAM signal.
- 16-5.** Consider the design of a power of  $N$  carrier recovery loop for a 4-PSK signal with  $|A_k| = 1$ . Suppose that  $\omega_c = 2\pi \times 2400 \text{ Hz} \pm 2\%$ .
- For the PLL in Figure 16-8, find  $\omega_n$  that leads to the smallest required lock range for the PLL. What is the lock range (the range of frequency offset  $\omega_0$  at the input to the phase detector)?
  - Suppose you wish to design the PLL with  $\omega_n = \omega_c = 2\pi \times 2400$ . Modify the design in Figure 16-8 so that this will work.

16-6. Consider the passband PAM signal

$$R(t) = \operatorname{Re}\{e^{j(\omega_c t + \theta(t))} S(t)\} \quad (16.25)$$

where

$$S(t) = \sum_{m=-\infty}^{\infty} A_m p(t - mT). \quad (16.26)$$

(a) Show that

$$R^2(t) = \frac{1}{2} |S(t)|^2 + \frac{1}{2} \cos(2\omega_c t + 2\theta(t)) \operatorname{Re}\{S^2(t)\} \quad (16.27)$$

- (b) Show that for 4-PSK  $E[R^2(t)]$  has no periodicity at the carrier frequency or multiples of the carrier frequency. Hence power of 2 carrier recovery will not be a good choice for 4-PSK.
- (c) Find conditions on  $A_k$  such that power of 2 carrier recovery will work, assuming that  $A_k$  is white.

## REFERENCES

1. L. E. Franks, "Synchronization Subsystems: Analysis and Design," in *Digital Communications: Satellite/Earth Station Engineering*, Prentice-Hall Inc. (1981).
2. D. D. Falconer, "Jointly Adaptive Equalization and Carrier Recovery in Two-Dimensional Digital Communication Systems," *BSTJ* 55(3)(March 1976).
3. W. C. Lindsey and M. K. Simon, "Data-Aided Carrier Tracking Loop," *IEEE Trans. Communications* COM-19(4) pp. 157-168 (April 1971).
4. W. C. Lindsey and M. K. Simon, "Carrier Synchronization and Detection of Polyphase Signals," *IEEE Trans. Communications* COM-20(2) pp. 441-454 (Feb. 1972).
5. R. Matyas and P. J. McLane, "Decision-Aided Tracking Loops for Channels with Phase Jitter and Intersymbol Interference," *IEEE Trans. Communications* COM-22(8) pp. 1014-1023 (Aug. 1974).
6. M. K. Simon and J. K. Smith, "Offset Quadrature Communications with Decision-Feedback Carrier Synchronization," *IEEE Trans. Communications* COM-22(10)(Oct. 1974).
7. U. Mengali, "Joint Phase and Timing Acquisition in Data Transmission," *IEEE Trans. Communications* COM-25(10) pp. 1174-1185 (Oct. 1977).
8. H. Kurihara, R. Katoh, H. Komizo, and H. Nakamura, "Carrier Recovery Circuit with Low Cycle Skipping Rate for CPSK/TDMA Systems," *Proc. 5-th Int. Conf. Digital Satellite Communication*, (March 1981).
9. M. Oerder, G. Ascheid, R. Hab, and H. Meyr, "An All Digital Implementation of a Receiver for Bandwidth Efficient Communication," in *Signal Processing III: Theories and Applications*, Elsevier Science Publishers B. V. (North-Holland) (1986).
10. R. D. Giulini, "Adaptive Phase-Jitter Tracker," *United States Patent, Patent No. 4,320,526*, (March 16, 1982).
11. R. P. Gooch and M. J. Ready, "An Adaptive Phase Lock Loop for Phase Jitter Tracking," *Proceedings of the 21st Asilomar Conf. on Signals, Systems, and Computers*, (Nov. 2-4, 1987).

# 17

---

## TIMING RECOVERY

---

The purpose of timing recovery is to recover a clock at the symbol rate or a multiple of the symbol rate from the modulated waveform. This clock is required to convert the continuous-time received signal into a discrete-time sequence of data symbols.

Many digital systems transmit a clock separate from the data stream. This is commonly done in systems which range from the size of an integrated circuit up to perhaps the size of a room. For digital communication systems, however, the transmission of a separate clock would be inefficient, since it requires additional facilities, bandwidth, or power. Hence, it is more economical to implement the additional circuitry which is required to derive the clock from the received modulated waveform itself. This is called *self-timing*, and requires that the timing information be implicit in the received waveform, which is not necessarily the case for all signaling methods.

### Example 17-1.

Consider a baseband system using an AMI line code (Section 12.1), in which "0" bits are transmitted as a zero voltage, and "1" bits are transmitted alternately as positive or negative-going pulses. If the user transmits a long sequence of zeros, the transmitted waveform will be identically zero, and there will be no timing information. □

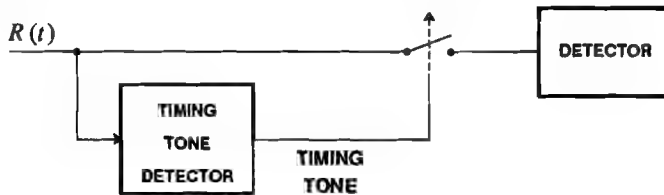
The need to do timing recovery imposes additional requirements on the modulation technique which are not present where a separate clock is available. The strength of the timing information in a signal is affected by the statistics of the signal, the line code, and the pulse shape.

**Example 17-2.**

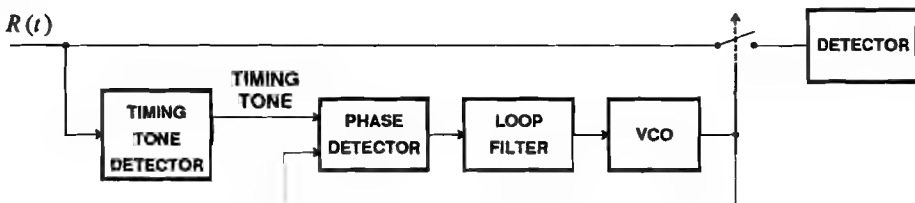
To correct the problem in the previous example, we can use a scrambler to randomize the data bits (see Section 12.5). In effect, we alter the statistics of the signal to ensure that a timing signal can be extracted at the receiver. The user can still foil the system by transmitting just the right bit sequence so that the output of the scrambler is zero, but for most practical cases, the probability of this occurring is negligible.  $\square$

Practical timing recovery circuits cannot perfectly duplicate the clock used at the remote transmitter. The most basic requirement is that the average frequency of the derived timing must exactly equal the average frequency of the transmitted signal. Obviously, the receiver must only generate as many bits as were transmitted, over the long term. Although the average frequency of the derived timing must be exact, the timing signal usually has phase jitter, or *timing jitter*. Timing jitter is not a fundamental impairment, but can be reduced to any desired level. We will see that the only barrier is cost.

There are fundamentally two types of timing recovery techniques which we call *deductive* and *inductive*. Deductive timing recovery directly extracts from the incoming signal a *timing tone*, which has an average frequency exactly equal to the symbol rate. The timing tone is used to synchronize the receiver to the incoming digital information, as shown in Figure 17-1. If the timing tone has unacceptable jitter, that jitter can be reduced using a PLL, as shown in Figure 17-2. This PLL will usually be of the



**Figure 17-1.** In deductive timing recovery, a timing tone, which is a signal with average frequency is exactly equal to the symbol rate, is extracted from the data signal. Typically, the zero crossings of the timing tone are used to determine when to sample the data signal.



**Figure 17-2.** Timing jitter can be reduced using a PLL.

type considered in Example 15-1 rather than Example 15-2, in that its objective will be produce a single-frequency rather than to track the jitter.

*Inductive* timing recovery does not directly process the received signal to get a timing tone, but rather uses a feedback loop as shown in Figure 17-3. Inductive timing recovery uses a PLL not as an added optimization to reduce timing jitter, but rather as an integral part of the method. One obvious advantage of this method is that most of the timing recovery can be done digitally and in discrete-time. A disadvantage is that the sampling rate of the received signal may have to be higher than the symbol rate in order to be able to estimate the timing error (although we will see baud-rate techniques that work). The phase detector in Figure 17-3 is the *sampling phase detector* of Figure 15-12.

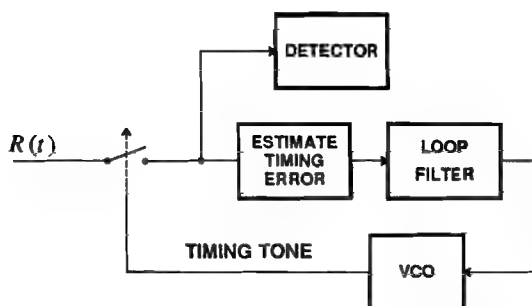
First we establish the criteria by which we will evaluate timing recovery techniques, and then discuss the most popular deductive timing recovery technique, the *spectral-line method*. After this, we describe an inductive technique, *MMSE timing recovery* and approximations. Finally, we describe a class of inductive techniques, *baud-rate timing recovery*, that allows sampling at the symbol-rate.

## 17.1. TIMING RECOVERY PERFORMANCE

Comparing the performance of alternative timing recovery methods analytically is usually very difficult. Some of the criteria for performance are discussed in this section.

### 17.1.1. Timing Phase

It is necessary to know not only how often to sample the data bearing signal, but also where to sample it. The choice of sampling instant is called the *timing phase*. For some signal schemes and some receivers, the performance of the receiver is



**Figure 17-3.** In inductive timing recovery, a current estimate of the timing tone is used to sample the signal. Then the timing error is estimated and the timing tone estimate is updated. This is effectively a PLL.

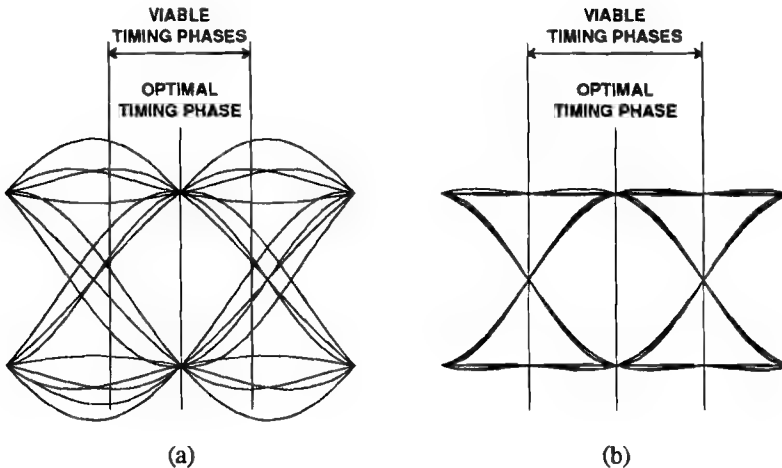
critically dependent on the timing phase. The sensitivity to timing phase can be quantified by examining the eye diagram of a signal (see Section 6.3). In Figure 17-4 we show eye diagrams for 25% and 100% excess bandwidth raised cosine binary antipodal PAM signaling. Clearly, the 25% excess bandwidth signal is more sensitive to timing phase because the eye closes more rapidly as the timing phase deviates from the optimum. With zero excess bandwidth, the signal is infinitely sensitive to timing phase, because the horizontal eye opening becomes zero (see Problem 6-5). It should not be inferred, however, that all signals without excess bandwidth have zero eye width. It is possible, using partial response line coding, to construct a signal with no excess bandwidth and an open eye. In effect, through coding we can disallow the sequences  $A_m$  that lead to large ISI.

### Example 17-3.

Modified duobinary line coding can be used to get a signal with zero excess bandwidth and an open eye (see Section 12.3). The eye width has been computed by Kabal and Pasupathy [1] and is about 36% of the symbol interval for zero excess bandwidth. To understand intuitively how this can be so, note that a zero excess bandwidth modified duobinary pulse  $p_{mdb}(t)$  is formed by subtracting two sinc pulses  $p_0(t)$ ,

$$p_{mdb}(t) = p_0(t) - p_0(t - 2T), \quad (17.1)$$

as shown in Figure 12-16b. The tails of the two sinc pulses tend to cancel each other, so the influence that the MDB pulse can have on distant symbols is considerably less than the influence that either sinc pulse alone would have.  $\square$



**Figure 17-4.** Eye diagrams for (a) 25% and (b) 100% excess bandwidth raised cosine pulses. The vertical lines indicate the range of possible timing phases (sample points) such that positive and negative pulses can be distinguished. In each case, the optimal timing phase is in the center where the eye opening is greatest.

Practical signals with zero excess bandwidth have an important advantage for timing recovery. Zero excess bandwidth signals are bandlimited to half the symbol rate frequency, so no aliasing occurs in sampling at the symbol rate. Even more remarkably, such a signal is *entirely insensitive* to timing phase if an adaptive equalizer (with enough taps) is used (see Problem 17-1). Thus, a fractionally-spaced equalizer (Section 11.5) is not required.

### 17.1.2. Timing Jitter

The timing tone recovered from a data signal always has timing jitter. This jitter can be reduced to any desired level by the design of timing recovery circuits, or by use of an additional phase-locked loop. Timing jitter introduces two degradations. First, the PAM signal is sampled at a sub-optimal point in the eye, increasing the ISI and thereby reducing noise immunity. Second, the bit stream emerging from the detector will generally have the same timing jitter produced by the timing recovery circuit. Usually, this is not a problem since consumers of the data (such as terminals or computers) are tolerant of limited amounts of jitter. In PCM transmission of a digital representation of a continuous-time signal such as speech, however, the jitter will cause the reconstructed speech to have non-uniform sampling, resulting in distortion. Timing jitter can have another serious consequence. For long distance transmission, a signal is often transmitted through a chain of many repeaters (Chapter 1). Each repeater reconstructs the digital signal and retransmits it, including the timing jitter on the input as well as timing jitter introduced in the repeater itself. The accumulation of timing jitter after a number of repeaters must be accounted for in the design of the timing recovery circuits (Section 17.5).

## 17.2. SPECTRAL-LINE METHODS

A baseband PAM signal carrying discrete-time digital information

$$R(t) = \sum_{k=-\infty}^{\infty} A_k p(t - kT) \quad (17.2)$$

is not stationary. In fact, it is *cyclostationary*, meaning that its moments vary in time and are periodic with period  $T$ , the symbol interval. Consider the new random process

$$Z(t) = f(R(t)) \quad (17.3)$$

where  $f(\cdot)$  is a memoryless nonlinearity. Often we will find that the mean-value of  $Z(t)$ ,  $E[Z(t)]$ , is non-zero and periodic with period  $T$ . We can think of this mean-value as a deterministic component of the random process  $Z(t)$ , consisting of a fundamental at the baud rate and harmonics. We can exploit this fact for timing recovery by forming  $Z(t)$  and passing it through a bandpass filter centered at the baud rate. This is known as *spectral-line timing recovery*.

If the mean-value of  $R(t)$  (corresponding to the identity function  $f(x) = x$ ) contains a spectral line at the baud rate, we can simply pass  $R(t)$  itself through a bandpass



filter to obtain a timing waveform. This is known as the *linear spectral-line method*, and it is discussed in Section 17.2.1. Most often, however, the mean-value is zero but higher moments of  $R(t)$  are periodic. When  $f(t)$  is nonlinear, this is called the *non-linear spectral-line method*, and is discussed in Section 17.2.2.

### 17.2.1. The Linear Spectral Line Method

When the mean-value of the data symbols is non-zero, the baseband PAM waveform may contain a spectral line at the baud rate. To determine whether a spectral line at the symbol frequency exists for a particular signal (17.2), partition the signal into a data independent (deterministic) component and a data dependent, zero mean stochastic component

$$R(t) = E[A_k] \sum_{m=-\infty}^{\infty} p(t - mT) + \sum_{m=-\infty}^{\infty} (A_m - E[A_k])p(t - mT). \quad (17.4)$$

The first term is independent of the data  $A_k$ , is periodic with period  $T$ , and can be thought of as a deterministic timing signal. Its periodicity implies a fundamental at the symbol rate  $2\pi/T$  and harmonics. If this fundamental has non-zero amplitude, it can be extracted with a bandpass filter, producing a timing tone. The second (data dependent) term is zero-mean and random, and results in jitter on the recovered timing tone.

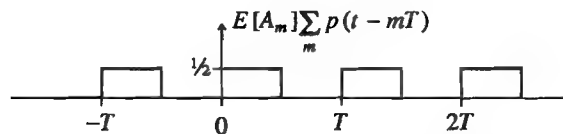
#### Example 17-4.

Consider a binary on-off signal with alphabet  $A_k \in \Omega_A = \{0,1\}$  in (17.2). Let the received pulse shape be a 50% duty cycle square pulse,

$$p(t) = \begin{cases} 1 & \text{for } 0 < t < T/2 \\ 0 & \text{otherwise} \end{cases} \quad (17.5)$$

Using this pulse shape, and observing that  $E[A_k] = 1/2$ , the deterministic part of (17.4) is shown in Figure 17-5. Clearly this signal has a strong spectral component at the symbol frequency.  $\square$

Even when the mean-value of  $R(t)$  is non-zero, it does not contain a spectral line at the baud rate unless the excess bandwidth is at least 100%.



**Figure 17-5.** The deterministic part of a binary on-off signal when the pulse shape is a 50% duty cycle square pulse.

**Exercise 17-1.**

Define  $x(t)$  to be the deterministic term of (17.4),

$$x(t) = E[A_k] \sum_{k=-\infty}^{\infty} p(t - kT). \quad (17.6)$$

Show that the Fourier transform  $P(j\omega)$  of the pulse must be nonzero at  $\omega = \pm 2\pi/T$  in order for  $X(j\omega)$  to have a component at  $\omega = \pm 2\pi/T$ . Hint: The results of appendix 15-A are useful.  $\square$

The requirement that the data symbols  $A_k$  have a non-zero mean (which implies a power penalty, Section 6.5), and that the excess bandwidth be at least 100%, usually rules out the linear spectral line method.

**17.2.2. The Nonlinear Spectral Line Method**

Often, even when the mean-value of  $R(t)$  is zero, the second and higher moments are non-zero and periodic. In 1958, in his classic paper on self-timing, Bennett observed that this can be exploited by passing the received signal through a memoryless nonlinearity [2]. The resulting waveform often has a deterministic mean-value that is periodic in the symbol rate, and a timing tone can be derived from it using a bandpass filter. To illustrate this, we will use the magnitude-squared nonlinearity  $f(x) = |x|^2$ . Assume a baseband PAM waveform of the form of (17.2), and for generality assume that  $A_k$  and  $p(t)$  are both complex-valued. The magnitude squared of this process depends on the correlation function of the data symbols, so assume they are white,

$$E[A_m A_n^*] = \sigma_A^2 \delta_{m-n}. \quad (17.7)$$

This assumption is reasonable, particularly when a scrambler (Section 12.5) is used. It is then straightforward to show that

$$E[|R(t)|^2] = \sigma_A^2 \sum_{m=-\infty}^{\infty} |p(t - mT)|^2. \quad (17.8)$$

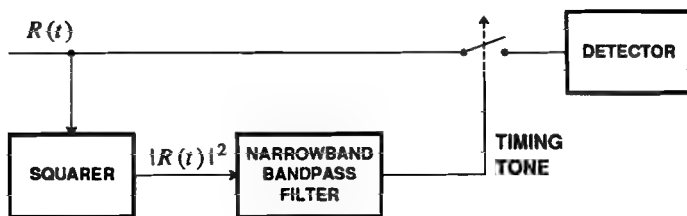
This expected value can again be considered a deterministic component of the process  $|R(t)|^2$ , and it is obviously periodic with period  $T$ , the symbol rate. The fundamental of this signal, if it has non-zero amplitude, will be extracted by the bandpass filter in Figure 17-6 to yield a timing tone at the symbol frequency. Some of the random component in  $|R(t)|^2$  will pass through the bandpass filter and result in timing jitter. In this case, any non-zero excess bandwidth is sufficient to guarantee a non-zero fundamental at the baud rate.

**Exercise 17-2.**

(a) Using the *Poisson's sum formula* (appendix 15-A), show that

$$\sum_{m=-\infty}^{\infty} |p(t - mT)|^2 = \frac{1}{T} \sum_{n=-\infty}^{\infty} Z_n e^{j2\pi n t / T}, \quad (17.9)$$

where



**Figure 17-6.** The squarer produces a deterministic periodic component, and the bandpass filter extracts the symbol frequency.

$$Z_n = \frac{1}{2\pi} \int_{-\infty}^{\infty} P(j\omega) P^*(j(\omega - n\frac{2\pi}{T})) d\omega. \quad (17.10)$$

Note that  $|Z_1| > 0$  as long as the bandwidth of  $P(j\omega)$  is greater than half the baud rate,  $\pi/T$ .

(b) Show that  $Z_{-n} = Z_n^*$  for all  $n$ .  $\square$

Usually  $p(t)$  has less than 100% excess bandwidth, in which case it is easy to show that the Fourier series coefficients are all zero except for the d.c. and the fundamental, and thus

$$E[|R(t)|^2] = Z_0 + 2\text{Re}\{Z_1 e^{j2\pi t/T}\}. \quad (17.11)$$

In this practical case the timing function contains no higher harmonics, although the bandpass filter is still required to reject as much as possible of the random portion of  $|R(t)|^2$  and any other noise or interference present.

#### Example 17-5.

Consider a received real-valued baseband signal formed with a binary antipodal signal constellation  $\Omega_A = \{\pm a\}$ . Since the signal is real-valued, the magnitude-squared signal is the same as the squared signal, and

$$R^2(t) = \sum_{m=-\infty}^{\infty} A_m^2 p^2(t - mT) + \sum_{\substack{m,n \\ m \neq n}} A_n A_m p(t - nT) p(t - mT). \quad (17.12)$$

Since  $A_m^2 = a^2$ , the first term is the deterministic timing tone, and the second term is the random portion that will be reflected in jitter at the bandpass filter output. There is no requirement that the data symbols have a non-zero mean, as in the linear spectral line method.  $\square$

It is generally desirable to have a larger timing tone, and hence large  $|Z_1|$ , since then the random components and noise will contribute relatively less to the timing jitter. Observe from (17.10) that the size of  $|Z_1|$  can generally be expected to increase as the amount of excess bandwidth increases, since there will be a larger overlap between  $P(j\omega)$  and  $P(j(\omega - 2\pi/T))$ . We conclude therefore that greater excess bandwidth is generally favorable, and doubly so when coupled with the fact that the

eye usually becomes less sensitive to timing jitter (wider eye) with larger excess bandwidth.

The analysis of the timing jitter is a little tricky. It may be tempting to compare the power in the deterministic timing tone to the power in the remaining random component that gets through the narrowband filter. We could find the power spectrum of the random part using the results of appendix 3-A. Appendix 3-A assumes a random phase epoch in order to get a WSS random process. In timing recovery, we are exploiting precisely the fact that the PAM signal is not WSS to derive the timing tone, and assuming wide-sense stationarity to determine timing jitter leads to significant inaccuracies. Accurate techniques have been developed for assessing the amount of timing jitter compared to the strength of the timing tone [3].

So far we have considered only a magnitude-squared nonlinearity, but there are other possibilities. For some signals, particularly when the excess bandwidth is low, a fourth-power nonlinearity  $|R(t)|^4$  is better than the magnitude-squared. In fact, fourth-power timing recovery can even extract timing tones from signals with zero excess bandwidth. An alternative nonlinearity is the magnitude  $|R(t)|$ , which for a real-valued signal is easily implemented as a full-wave rectifier. Although the analysis is more difficult, simulations show that absolute-value circuits usually outperform square-law circuits for signals with low excess bandwidth [4]. Like fourth-power circuits, absolute-value circuits can extract a timing tone for signals with zero excess bandwidth. For signals with low excess bandwidth, however, a fourth-power timing circuit may be preferable. Simulations for QPSK [4] suggest that fourth-power circuits outperform absolute-value circuits for signals with less than about 20% excess bandwidth.

If timing recovery is done in discrete-time, aliasing must be considered in the choice of a nonlinearity. Any nonlinearity will increase the bandwidth of the PAM signal, and in continuous time this is not a consideration since the bandpass filter will reject the higher frequency components. In the presence of sampling, however, the high frequency components due to the nonlinearity can alias back into the bandwidth of the bandpass filter, resulting in additional timing jitter. This is obviously true if the nonlinearity precedes the sampling, but is also true even if the nonlinearity is performed after sampling. This is because preceding a sampling operation with a memoryless nonlinearity is mathematically equivalent to reversing the order of the sampling and the nonlinearity. Therefore, in a discrete-time realization, a magnitude-squared nonlinearity usually has a considerable advantage over either absolute-value or fourth-power nonlinearity. In particular, raising a signal to the fourth-power will quadruple its bandwidth, and full-wave rectifying spreads its bandwidth even more. Each situation must be considered independently to determine whether the aliasing is detrimental.

The advantages of absolute-value and fourth-power circuits over square-law circuits can be at least partly compensated by *prefiltering* of the PAM signal prior to the nonlinearity. This prefilter can reduce the timing jitter substantially, particularly for low excess bandwidth. Prefiltering is based on the observation that often much of the spectrum of the PAM signal does not contribute to the timing tone, so a filter that eliminates the unnecessary part of the spectrum will reduce timing jitter. This is best

seen by example.

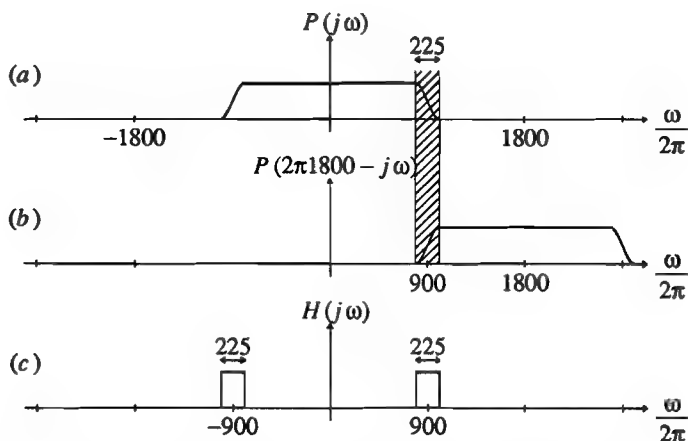
### Example 17-6.

Consider a real-valued baseband PAM signal with 12.5% excess bandwidth and a symbol rate of 1800 baud. We will show that the timing tone of a square law timing recovery circuit is not altered by *prefiltering* the signal using an ideal bandpass filter with passband from 787.5 Hz to 1012.5 Hz. Such prefiltering, however, removes much of the noise and signal components that would only contribute to the jitter. The strength of the timing tone will be determined by the Fourier series coefficient  $Z_1$ . This can be found using (17.10), and in particular

$$Z_1 = \frac{E[|A_k|^2]}{T} \int_{-\infty}^{\infty} P(j\frac{2\pi}{T} - j\omega) P(j\omega) d\omega. \quad (17.13)$$

$P(j\omega)$  is shown in Figure 17-7a. To get  $Z_1$ , the spectrum in Figure 17-7a is multiplied by the spectrum in Figure 17-7b and the product is integrated, per (17.13). Only the cross-hatched regions will contribute to  $Z_1$ , which will not change if the received signal is first filtered with the prefilter shown in Figure 17-7c. Since the receiver must work with  $|R(t)|^2$  and not its expected value, parts of the PAM signal that would contribute to the jitter and not to the timing tone have been removed, as well as undoubtedly some noise components. The bandwidth of the prefilter depends on the excess bandwidth of the pulse. Given 12.5% excess bandwidth, the ideal prefilter bandwidth is 225 Hz, as shown.  $\square$

Prefiltering is not as helpful for signals with large excess bandwidths (see Problem 17-5).



**Figure 17-7.** Illustration of the computation of the Fourier series coefficient  $Z_1$  for a baseband PAM signal with a pulse whose Fourier transform is shown in (a). To get  $Z_1$ , the function in (a) is multiplied term-wise by the function in (b) and the product is integrated. Only the cross-hatched regions will contribute to the answer. In (c), the prefilter shown applied to PAM signal prior to the square nonlinearity would not affect  $Z_1$  but would reduce jitter.

### 17.2.3. The Spectral Line Method for Passband Signals

For a baseband PAM system, the previous results apply directly. For a passband PAM system, since a complex-valued baseband signal was considered, they would also apply if demodulation was performed first. However, there are a couple of problems with this approach:

- As described in Chapter 16, demodulation is usually performed last, after timing recovery and equalization.
- Carrier recovery loops are typically decision-directed, and thus depend for their operation on availability a stable timing phase. If the timing recovery depends on carrier recovery, and vice versa, there might be serious problems in initial acquisition.

Fortunately, it is not necessary to demodulate before performing timing recovery. By deriving timing directly from a passband signal, we completely decouple timing recovery from other functions in the receiver. For the spectral-line method, a passband signal can be passed directly through a non-linearity such as a magnitude-square, and a timing tone that can be bandpass filtered will result. This technique is sometimes called *envelope derived timing*, because the squaring and filtering operation is similar to extracting the *envelope* of the signal.

The simplest case results when we have an analytic bandpass filter, or equivalently bandpass filter and phase splitter, at the front end of the receiver (Section 6.4). The output of this front end will be an analytic signal

$$Y(t) = R(t)e^{j\omega_c t} \quad (17.14)$$

where  $R(t)$  is a complex-valued baseband signal as given by (17.2). Since

$$|Y(t)| = |R(t)| \cdot |e^{j\omega_c t}| = |R(t)| \quad (17.15)$$

we arrive at the conclusion that applying the analytic signal to a magnitude, magnitude-square, or fourth-power nonlinearity is equivalent to applying the demodulated baseband signal.

However, there are some possible difficulties with deriving timing from the analytic signal.

#### Example 17-7.

Often the front-end of the receiver consists of a bandpass filter, sampler, and discrete-time phase splitter. The sampler would usually be controlled by the derived timing. Thus, the analytic signal depends on the timing recovery output through the sampling phase and frequency, and using this signal to derive timing may lead to undesirable interactions and acquisition problems.  $\square$

Fortunately, there is a simple alternative to using the analytic signal; namely, to apply just the real part  $\text{Re}\{Y(t)\}$  to the nonlinearity. Since  $\text{Re}\{Y(t)\}$  is the phase splitter input, and not output, this approach eliminates any problems encountered in Example 17-7. Considering again the squarer nonlinearity,

$$(\text{Re}\{Y(t)\})^2 = \frac{1}{2}|Y(t)|^2 + \frac{1}{4}Y^2(t) + \frac{1}{4}(Y^*(t))^2. \quad (17.16)$$

Under reasonable assumptions, the expected value of the last two terms in (17.16) is zero.

### Exercise 17-3.

Assume the zero-mean complex-valued data symbols  $A_k$  are independent of one another and have independent real and imaginary parts with equal variance. Show that  $E[A_m A_n] = 0$  for all  $m$  and  $n$ . (If this result looks strange, recall that the variance of the symbols is  $E[|A_k|^2]$  and not  $E[A_k^2]$ ).  $\square$

It follows directly from Exercise 17-3 that  $E[Y^2(t)] = 0$  under these same assumptions on the data-symbol statistics. Thus, we get that

$$E[(\operatorname{Re}\{Y(t)\})^2] = \frac{1}{2}E[|Y(t)|^2], \quad (17.17)$$

and the square of the real part of the analytic signal contains the same timing function (albeit half as large) as the magnitude-squared of the analytic signal. However, we do pay a price:

- The timing function is half as large, making sources of jitter more important.
- In (17.16), the second and third terms are additional inputs to the bandpass filter. These terms have zero mean, and hence do not affect the timing function. However, they do represent an additional source of randomness that contributes to a larger timing jitter.

The same relative merits of squaring, absolute-value, and fourth-power techniques apply to passband timing recovery as to baseband. In particular, absolute-value and fourth-power are usually better than squaring, except when aliasing is a problem in discrete-time implementations. As with baseband signals, it is sometimes advantageous to prefilter the signal before squaring.

## 17.3. MMSE TIMING RECOVERY AND APPROXIMATIONS

Although the spectral-line method is the most popular, it is not always suitable. It can be difficult to use when the timing recovery has to be done in discrete-time.

### Example 17-8.

When an echo canceler is used to separate the two directions in a full-duplex connection (Chapter 19), the inherently discrete-time echo cancellation must be performed prior to timing recovery. It is often convenient to avoid reconstruction of a continuous-time signal, and perform timing recovery in discrete time. The echo canceler complexity is proportion to the sampling rate (Chapter 19), and therefore there is motivation to minimize the sampling rate for the timing recovery function.  $\square$

In this section we describe *minimum mean-square error (MMSE) timing recovery*, a technique that is not practical in its exact formulation, but which has numerous practical approximations. MMSE timing recovery is also sometimes called *LMS timing recovery*, and is similar to *maximum likelihood* timing recovery [5]. The MMSE

family of techniques fit the general framework of Figure 17-3, so they are *inductive*. The phase detector is essentially the *sampling phase detector* of Figure 15-12, the main difference being that the input signal has a more complicated form than the simple sinusoid assumed in Chapter 15.

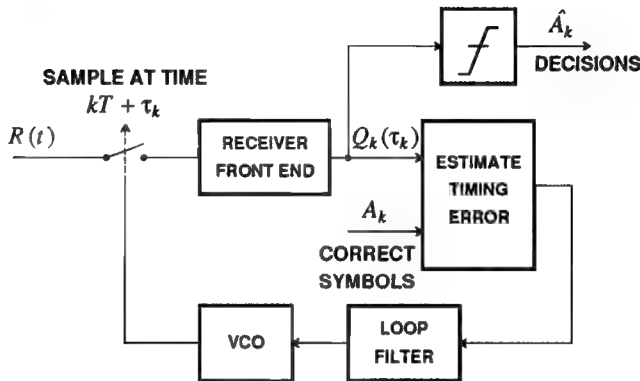
### 17.3.1. The Stochastic Gradient Algorithm

In Figure 17-8, the received signal (assumed baseband) is sampled at times  $(kT + \tau_k)$ . The symbol interval is  $T$ , and thus  $\tau_k$  represents the timing error in the  $k$ -th sample. After some possible front-end processing, the notation for the  $k$ -th sample is  $Q_k(\tau_k)$ , rather than just  $Q_k$ , to emphasize the dependence on the timing phase. Usually,  $\tau_k$  is determined by zero crossings of the timing tone (see Problem 17-6). Ideally,  $\tau_k$  is a constant corresponding to the best sampling phase, but in practice  $\tau_k$  has timing jitter. Inductive timing recovery is best understood as a technique for iteratively adjusting  $\tau_k$ .

MMSE timing recovery adjusts  $\tau_k$  to minimize the expected squared error between the input to the slicer and the correct symbol,

$$E[|E_k(\tau_k)|^2] = E[|Q_k(\tau_k) - A_k|^2], \quad (17.18)$$

with respect to the timing phase  $\tau_k$ , where  $A_k$  is the correct data symbol. This is the same criterion used for adaptive equalizers in Chapter 11. Just as with adaptive equalizers, MMSE timing recovery can use the stochastic gradient algorithm to try to find the optimal timing phase. If correct data symbols  $A_k$  are available at the receiver (for example during a training sequence at system startup), the structure is shown in Figure 17-8. This criterion directly minimizes the slicer error, and hence should result in close to the minimum error probability. Unfortunately, the input to the slicer  $Q_k(\tau_k)$  is a complicated non-linear function of the timing phase  $\tau_k$ , so unlike the adaptive



**Figure 17-8.** MMSE timing recovery adjusts the timing phase to minimize the squared error between the input to the slicer and the correct symbols.



equalizer case there may not be a well-defined unique minimum MSE timing phase. In addition, finding an analytic closed-form solution may be impossible.

Instead of seeking a closed-form solution, we can try to minimize the expected squared error by adjusting the timing phase in the direction opposite the derivative of the expected value of the squared error,

$$\frac{\partial}{\partial \tau_k} E[|E_k(\tau_k)|^2] = E \left[ \frac{\partial}{\partial \tau_k} |E_k(\tau_k)|^2 \right]. \quad (17.19)$$

**Exercise 17-4.**

Show that for any complex function  $E_k(\tau_k)$  of the real variable  $\tau_k$ ,

$$\frac{\partial}{\partial \tau_k} |E_k(\tau_k)|^2 = 2 \operatorname{Re} \left\{ E_k^*(\tau_k) \frac{\partial E_k(\tau_k)}{\partial \tau_k} \right\}. \quad (17.20)$$

□

Since  $A_k$  does not depend on  $\tau_k$  we can write

$$\frac{\partial E_k(\tau_k)}{\partial \tau_k} = \frac{\partial Q_k(\tau_k)}{\partial \tau_k}. \quad (17.21)$$

Hence, to adjust the timing phase in the direction opposite the gradient, we use

$$\tau_{k+1} = \tau_k - \alpha \operatorname{Re} \left\{ E_k^*(\tau_k) \frac{\partial Q_k(\tau_k)}{\partial \tau_k} \right\}. \quad (17.22)$$

We have dispensed with the expectation in (17.19), so this is a *stochastic gradient algorithm*. The *step size*  $\alpha$  is usually determined empirically to ensure stability, minimize timing jitter, and ensure adequate tracking ability.

The signal  $Q_k(\tau_k)$  consists of samples of some continuous-time signal  $Q(t)$  taken at  $t = kT + \tau_k$ , so

$$\frac{\partial}{\partial \tau_k} Q_k(\tau_k) = \left[ \frac{\partial}{\partial t} Q(t) \right]_{t=kT+\tau_k}. \quad (17.23)$$

Hence (17.22) is equivalent to

$$\begin{aligned} \tau_{k+1} &= \tau_k - \alpha \operatorname{Re} \left\{ E_k^*(\tau_k) \left[ \frac{\partial}{\partial t} Q(t) \right]_{t=kT+\tau_k} \right\} \\ &= \tau_k - \alpha \operatorname{Re} \left\{ [Q_k(\tau_k) - A_k]^* \left[ \frac{\partial}{\partial t} Q(t) \right]_{t=kT+\tau_k} \right\}. \end{aligned} \quad (17.24)$$

This update is shown in Figure 17-9.

As with equalization, there would normally be an acquisition or training period during which the symbols  $A_k$  are available, followed by a decision-directed tracking period during which the actual symbols  $A_k$  are replaced by decisions  $\hat{A}_k$ . Furthermore, since  $E_k(\tau_k)$  is not a linear function of  $\tau_k$ ,  $|E_k(\tau_k)|^2$  is not a quadratic function

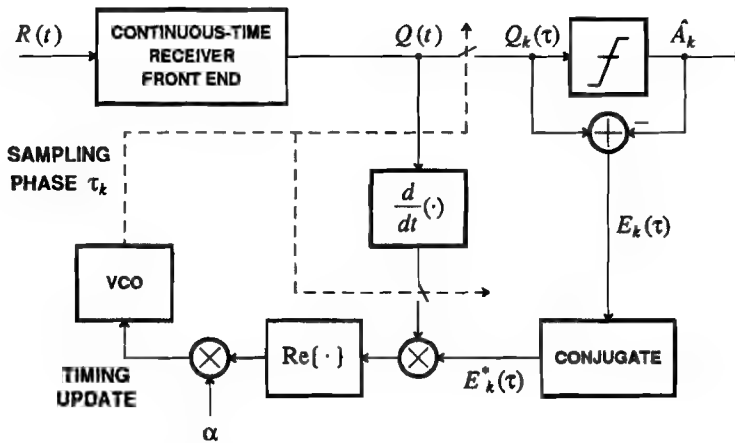
of  $\tau_k$ , and the stochastic gradient algorithm is not guaranteed to converge to the optimal timing phase. In practice, some other technique should be used during acquisition to get a reasonable initial estimate of the correct timing phase [6]. This has the disadvantage of requiring two different timing recovery techniques during acquisition and tracking.

The method of (17.24) still doesn't accomplish the basic objective of using only *samples* of the received signal, since the receiver front end up to the slicer must be implemented in continuous-time. One approach is to differentiate the continuous-time received signal  $R(t)$ , sampling this derivative and processing it with a replica of the receiver front end [5], but this approach requires *two* receiver front ends and is often incompatible with echo cancellation (Chapter 19). More approximations are needed to eliminate the need for a continuous-time version of the slicer input, as discussed in the next subsection.

The stochastic gradient timing recovery operates on the input to slicer, which has been equalized and demodulated. It therefore relies on the convergence of the equalizer (Chapter 11) and, for passband PAM, the acquisition of carrier (Chapter 16). The equalizer and carrier recovery themselves usually assume that the timing recovery has acquired! This potentially leads to a serious interaction among these three functions. This problem is overcome at least partially if the timing phase is estimated at startup by some other method not requiring equalizer convergence [6]. More recently [7] the interaction with carrier acquisition in passband systems has been eliminated by modifying the error criterion to

$$E_k = |Q_k(\tau_k)|^2 - |\hat{A}_k|^2. \quad (17.25)$$

For passband PAM signals, an incorrect carrier phase will only *rotate* the received



**Figure 17-9.** Stochastic gradient timing recovery using a continuous-time version of the input to the slicer.

signal  $Q_k(\tau_k)$  in the complex plane; its magnitude squared is not affected. Hence timing recovery based on this error criterion will not be sensitive to carrier phase.

### 17.3.2. Other Approximate MMSE Techniques

The need for a continuous-time version of the slicer input,  $Q(t)$ , can be avoided by one of two techniques. The first, published by Qureshi in 1976, assumes that Nyquist-rate samples of  $Q(t)$  are available [6]. Since the derivative is an LTI system, it can be realized as a discrete-time filter. In appendix 15-B we show that if  $Q(t)$  is sampled at the Nyquist rate, then

$$\frac{\partial Q_k(\tau_k)}{\partial \tau_k} = Q_k(\tau_k) * d_k \quad (17.26)$$

where  $d_k$  is given by (17.51). It is necessary to assume that  $\tau_k$  varies slowly for this to be valid (see the appendix). This suggests that if the received signal is sampled at the Nyquist rate, timing recovery can be implemented as shown in Figure 17-10. The derivative is computed by a discrete-time filter with impulse response  $d_k$ , which for practical reasons is approximated by an FIR filter. A particularly simple approximation is given by (17.52), yielding

$$\tau_{k+1} = \tau_k + \alpha Z_k(\tau_k) \quad (17.27)$$

where

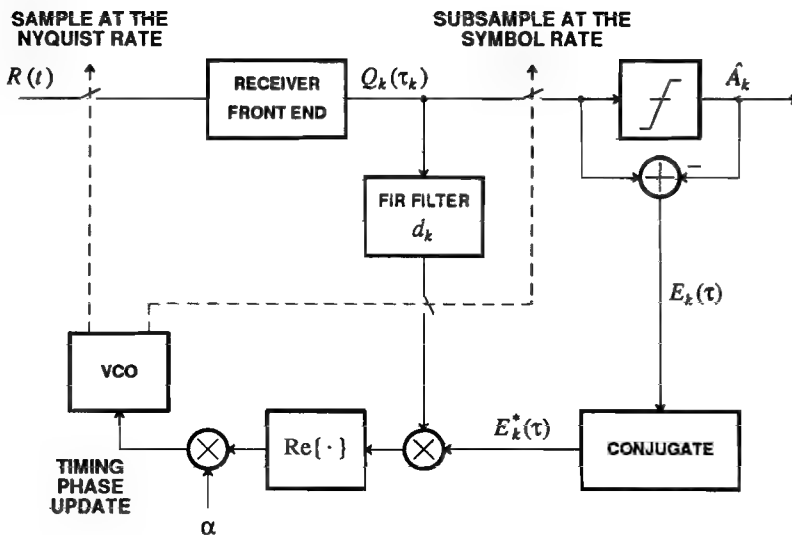


Figure 17-10. Stochastic gradient timing recovery when the received signal  $R(t)$  is sampled at the Nyquist rate.

$$Z_k(\tau_k) = -\operatorname{Re}\left\{ [Q_k(\tau_k) - \hat{A}_k][Q_{k+1}(\tau_k) - Q_{k-1}(\tau_k)] \right\}. \quad (17.28)$$

The assumption of Nyquist sampling often complicates the receiver front end, which may otherwise get away with a lower sampling rate. There are exceptions to this.

#### Example 17-9.

Using partial response line coding (Section 12.3), it is possible to use zero excess bandwidth. In this case, the symbol rate is the Nyquist rate, and this method imposes no computational penalty.  $\square$

#### Example 17-10.

It is common to use fractionally-spaced equalizers (Section 11.5) so that the receiver is relatively insensitive to timing phase. In this case, the equalizer itself may operate on Nyquist-rate samples. Hence all parts of the receiver that precede the equalizer must operate on Nyquist-rate samples. However, there is still a penalty in complexity for the technique in Figure 17-10. Usually, the output of a fractionally spaced equalizer is immediately decimated down to the symbol rate. This means that the output of the equalizer needs only to be computed at the symbol rate. To use this timing scheme, we would have to compute the output of the equalizer at the Nyquist rate.  $\square$

A second approach to avoiding a continuous-time front-end to the receiver has recently been explored [7]. From (17.64), it is sufficient to directly compute the derivative of the *error signal* with respect to timing phase. This derivative can be approximated by taking each sample either slightly ahead or slightly behind the current estimate  $\tau_k$  of the timing phase. The two phases are alternated, and the difference between the error at even samples and the error at odd samples is an indication of the derivative of the error with respect to timing phase.

### 17.3.3. Approximate Maximum-Likelihood Methods

Many *ad hoc* timing recovery techniques have appeared over time, mostly justified first heuristically and then empirically. Several of these have been shown to be approximations to a *maximum likelihood* (ML) timing recovery [5,8], and bear a strong resemblance to MMSE methods.

One such method, called the *sample-derivative method*, is shown in Figure 17-11 [9] for a baseband PAM system. This technique can be justified heuristically by observing that it will attempt to move the sampling phase until the derivative is zero, which occurs at the peaks of the signal, presumably a good (but not optimal) place to sample. For discrete time systems, approximations to the derivative (appendix 15-B) are used.

A related technique, shown in Figure 17-12, is called the *early-late gate method* [8]. In this technique, the received waveform is sampled two extra times, once prior to the sampling instant by  $\Delta/2$  and once after the sampling instant by the same amount. The sampling instant is adjusted until the two extra samples are equal.

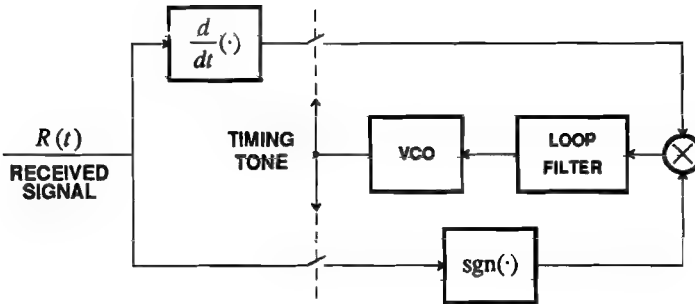


Figure 17-11. Sample-derivative timing recovery. The box labeled  $\text{sgn}(\cdot)$  computes the signum of the input, and can be implemented as a hard limiter.

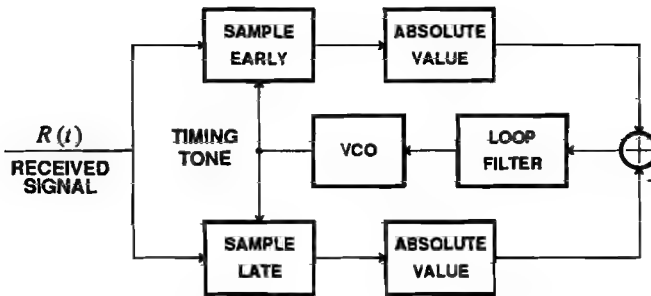


Figure 17-12. Early-late gate timing recovery.

## 17.4. BAUD-RATE TIMING RECOVERY

Interest in realizing timing recovery using symbol-rate sampling, especially in conjunction with echo cancellation (Chapter 19), has led to interest in a class of techniques known as *baud-rate timing recovery*. Interestingly, the timing phase update given by (17.27) works even when the samples are taken at the symbol rate and not at the Nyquist rate! First we will justify this claim, and then describe a family of related baud-rate techniques that are similar to but better than (17.27). Any of the baud-rate techniques derived in this section can be used in the receiver configuration shown in Figure 6-23. An additional frequency synthesizing PLL may be required to generate the multiple sampling frequencies.

Define the *timing function* to be the expected timing update, given the current timing phase  $\tau_k$ ,

$$f(\tau_k) = E[Z_k(\tau_k)] . \quad (17.29)$$

From (17.28),

$$f(\tau_k) = -\operatorname{Re}\{E[Q_k(\tau_k)Q_{k+1}(\tau_k)] - E[Q_k(\tau_k)Q_{k-1}(\tau_k)] \\ - E[\hat{A}_k Q_{k+1}(\tau_k)] + E[\hat{A}_k Q_{k-1}(\tau_k)]\}. \quad (17.30)$$

If  $Q_k(\tau_k)$  is WSS random process, and  $\tau_k$  varies insignificantly from one sample to the next, then the first two terms are equal, and

$$f(\tau_k) = \operatorname{Re}\left\{E[\hat{A}_k Q_{k+1}(\tau_k)] - E[\hat{A}_k Q_{k-1}(\tau_k)]\right\}. \quad (17.31)$$

We can show, under benign assumptions for a baseband PAM signal, that  $Q_k(\tau_k)$  is WSS.

#### Exercise 17-5.

Write the input to the slicer as samples of a continuous-time PAM signal, with the  $k^{\text{th}}$  sample taken at time  $t = kT + \tau_k$ ,

$$Q_k(\tau_k) = \sum_{m=-\infty}^{\infty} A_m p((k-m)T + \tau_k) + N_k. \quad (17.32)$$

- (a) Assume that  $A_k$  and  $N_k$  in (17.32) are WSS random processes, and that  $N_k$  is zero mean and independent of  $A_k$ . Show that  $Q_k(\tau_k)$  is WSS. Show that under these assumptions, and also assuming that  $Q_k(\tau_k)$  is real-valued, the first two terms in (17.30) are equal and hence cancel.  $\square$

#### Exercise 17-6.

Assume that all the decisions are correct,  $\hat{A}_k = A_k$ , that  $A_k$  is white, and that  $p(t)$  is real. Show that the timing function is

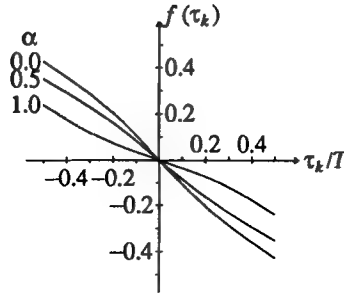
$$f(\tau_k) = E[|A_k|^2][p(\tau_k + T) - p(\tau_k - T)]. \quad (17.33)$$

$\square$

If  $p(t)$  is symmetric about  $\tau_k$ , the timing function will be zero at  $f_k(0) = 0$ . This is the condition under which the *average* timing phase update in (17.27) is zero, and hence the point to which the timing recovery algorithm will converge. In the case of a symmetric  $p(t)$  this is also a good timing phase.

For a given pulse  $p(t)$ , it is a good idea to check the timing function in (17.33) to ensure that it is monotonic and has a unique zero-crossing, and that this zero-crossing is at a good timing phase. Equation (17.33) is plotted in Figure 17-13 for raised cosine pulses with various excess bandwidths. Notice that if  $\tau_k$  is early, the timing function is positive, so the expected timing phase update is positive, which is what we want.

The timing function was derived without assuming Nyquist-rate sampling, so this technique should work with symbol-rate samples. However, it can be improved. In Exercise 17-5 we showed that the first two terms in (17.30) cancel. However, since the timing phase update is  $Z_k$  not  $E[Z_k]$ , these two terms will contribute to the timing jitter. A closely related baud-rate technique that does not have these two terms was given by Mueller and Müller in 1976 [10]. In fact, they gave a general technique that



**Figure 17-13.** The timing function  $f(\tau_k)$  is the expected value (over all possible symbol sequences) of the timing phase update as a function of the timing phase  $\tau_k$ . It is shown here for the timing phase update in (17.28), for three raised-cosine pulse shapes with rolloff factors  $\alpha$ . Notice that in each case it has a unique zero crossing at the optimal timing phase, and that the polarity is correct to ensure that the timing phase will tend to be adjusted towards the zero crossing. (After Mueller and Müller [10]).

can yield a variety of timing functions, of which (17.33) is only one. Another suitable one is

$$f(\tau_k) = E[|A_k|^2]p(\tau_k + T). \quad (17.34)$$

The general technique uses timing updates of the form

$$Z_k = \mathbf{X}_k' \mathbf{Q}_k \quad (17.35)$$

in (17.27), where  $\mathbf{Q}_k$  is a vector of the last  $m$  samples and  $\mathbf{X}_k$  is a (yet unspecified) vector function of the last  $m$  symbols

$$\mathbf{Q}_k = \begin{bmatrix} Q_{k-m+1}(\tau_k) \\ \vdots \\ Q_k(\tau_k) \end{bmatrix} \quad \mathbf{X}_k = \begin{bmatrix} x_1(A_{k-m+1}, \dots, A_k) \\ \vdots \\ x_m(A_{k-m+1}, \dots, A_k) \end{bmatrix}. \quad (17.36)$$

To get the timing function in (17.33), simply choose

$$\mathbf{X}_k = \begin{bmatrix} -A_k \\ A_{k-1} \end{bmatrix} \quad (17.37)$$

(see Problem 17-7). This technique will have less timing jitter than that given by (17.28). Of course, to make this decision directed, use  $\hat{A}_k$  instead of  $A_k$ .

## 17.5. ACCUMULATION OF TIMING JITTER

In digital transmission systems there are often long chains of regenerative repeaters. We saw in Chapter 1 how beneficial regeneration was in preventing the accumulation of noise and distortion through the transmission medium. However, the regenerative repeaters unfortunately allow an accumulation of timing jitter, which

can become a critical problem if it is not properly controlled through the careful design of the timing recovery circuits [11]. This accumulation of jitter will typically result in a limit in the number of repeaters of a given design before the accumulated jitter becomes intolerable.

When considering accumulation of jitter it is important to distinguish between two basic sources of jitter. Consider, for example, the spectral-line method of Figure 17-6. The timing recovery circuit responds to the timing function, which is the deterministic mean-value component of  $|R(t)|^2$ . The jitter arises from the random components of this signal. One random component is the random portion due to the PAM signal itself, and this is called the *systematic* or *data-dependent* jitter. The second component of jitter is due to any noise or crosstalk signals present in  $R(t)$ . This is called the *random* jitter.

In a long chain of repeaters, the systematic jitter greatly dominates the random jitter. This is because, in the absence of transmission errors, the systematic jitter component is the *same* in each and every repeater, and therefore adds coherently. A model for the accumulation of this type of jitter is shown in Figure 17-14. Any dependency of the jitter introduced in one repeater on the jitter introduced upstream will be slight as long as the total introduced jitter remains modest, and therefore we can assume that the jitter introduced in each repeater is additive. Each repeater has an equivalent transfer function to jitter,  $H(j\omega)$ .

#### Example 17-11.

$H(j\omega)$  is typically a low-pass filter response. When timing recovery is implemented using a PLL, the closed-loop response is a lowpass filter, as shown in Chapter 15. For the spectral-line methods, the output of the bandpass filter can have only frequencies in the vicinity of the baud rate, and thus the jitter components are also low frequency.  $\square$

The response  $H(j\omega)$  is applied not only to the jitter introduced in the same repeater, but also the jitter introduced in all repeaters upstream. The overall transfer function to jitter at the output is

$$H_{\text{TOTAL}}(j\omega) = \sum_{i=1}^N H^i(j\omega) = H(j\omega) \cdot \frac{1 - H^N(j\omega)}{1 - H(j\omega)} \quad (17.38)$$

for  $N$  repeaters. For a given  $H(j\omega)$  and jitter power spectrum, this relation allows us to predict the total jitter as a function of the number of repeaters.

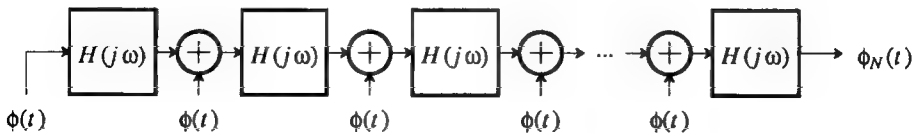


Figure 17-14. A model for the accumulation of systematic data-dependent jitter.



Since the bandwidth of the jitter transfer function is generally narrow relative to the baud rate, the jitter spectrum is not expected to vary substantially within the bandwidth of  $H(j\omega)$ , and it is a good approximation to assume that it has a white power spectrum  $\Phi_0$ . The power spectrum of the jitter at the output of the line is then  $\Phi_0 |H_{\text{TOTAL}}(j\omega)|^2$ , from which we can predict the total mean-square jitter. Our main concern is in how this jitter varies with  $N$ , since this accumulation of jitter may limit the number of allowable repeaters for a given timing recovery design.

It is instructive to consider three cases:

- At frequencies where  $|H(j\omega)| > 1$ , we can readily see that  $|H_{\text{TOTAL}}(j\omega)| \rightarrow \infty$  as  $N \rightarrow \infty$ . It is therefore a very bad idea to allow any jitter gain at any frequency for a repeatered line. This rules out any PLL with peaking, for example, many second-order PLLs.
- For frequencies where  $|H(j\omega)| \ll 1$ , we can see that  $H_{\text{TOTAL}} \rightarrow H/(1-H)$  as  $N \rightarrow \infty$ . Thus, at frequencies where  $|H(j\omega)|$  is close to zero there is no jitter accumulation, and the only significant jitter is that introduced in the last repeater.
- The critical case to examine is when  $|H(j\omega)| \approx 1$ , which will occur within the passband of the single-repeater jitter transfer function. A simple application of L'Hospital's rule establishes that  $|H_{\text{TOTAL}}(j\omega)| \rightarrow N$  as  $|H(j\omega)| \rightarrow 1$ . Thus, the jitter at frequencies where the jitter transfer function is precisely unity accumulates coherently as expected —  $N$  repeaters results in  $N$  times the jitter amplitude.

#### Example 17-12.

Assume the jitter transfer function of each repeater is an ideal lowpass filter with bandwidth  $\omega_1$ . Then the overall jitter power spectrum is  $\Phi_0 N^2$  within the passband and zero elsewhere, so the jitter power is  $\omega_1 \Phi_0 N^2 / \pi$ . Thus, the jitter power increases as the square of the number of repeaters, or the jitter amplitude increases in proportion to  $N$ . For any desired  $N$  we can limit the accumulated jitter by making the bandwidth of the lowpass filter small.  $\square$

#### Example 17-13.

A much more realistic assumption is that the jitter transfer function in each repeater is a single-pole lowpass filter, obtained, for example, by using a first order PLL with loop filter  $L(s) = K_L$ ,

$$H(j\omega) = \frac{K_L}{K_L + j\omega}, \quad (17.39)$$

in which case the accumulated jitter transfer function is

$$H_{\text{TOTAL}}(j\omega) = \frac{K_L}{j\omega} \left[ 1 - \frac{K_L^N}{(K_L + j\omega)^N} \right]. \quad (17.40)$$

The total jitter power in this case can be shown [12,11] after evaluation of a rather complicated integral to be  $K_L \Phi_0 N / \pi$ . Thus, the jitter amplitude has been reduced from an  $N$  dependence for an ideal lowpass filter (see Example 17-12) to a  $\sqrt{N}$  dependence. Even though the jitter transfer function is always  $N$  at very low frequencies, as  $N$  increases the bandwidth over which this dependence is valid narrows, thus slowing the rate of

accumulation of jitter (see Problem 17-9). In this case the departure of the lowpass filter from ideality is beneficial!  $\square$

For PCM transmission of continuous-time signals, the limit on accumulated timing jitter will be the distortion suffered due to the irregular spacing of the received samples. While repeated digital communications systems suffer from this problem of jitter accumulation, it is not a fundamental impairment. To reduce the jitter to any desired level, we can pass the bit stream through a timing recovery circuit with a bandwidth much smaller than the bandwidth of the regenerator timing circuits.

## 17.6. FURTHER READING

Tutorial articles on timing recovery are rare, perhaps because the ideas are subtle and the analysis is relatively difficult. One of the few such articles is by Franks [3]. Another basic reference was written by Gitlin and Hayes [13]. The classic article by Bennett, published in 1958 [2], is well worth reading because of its fine craftsmanship and historical value. Another classic article with a lucid discussion of the topic is written by Aaron [14].

The relationship between fractionally-spaced equalizers and timing recovery has been extensively studied [15,16]. Baud-rate timing recovery is described by Mueller and Müller [10] and others [17,18]. The effect of timing jitter on echo cancelers has been examined [19,20,21]. Even the oldest techniques are still being pursued, as for example in [22] where block codes that permit linear timing recovery are studied. Of historical interest is one of the earliest discussions of self-timing by Sunde [23], in which a linear spectral-line method is proposed. Discrete-time timing recovery for voiceband data modems is described in [24]. Other papers with interesting techniques or analysis are [25,9].

## APPENDIX 17-A THE POISSON SUM FORMULA

In the analysis of the spectral-line method, we need to relate the Fourier series of a periodic signal in summation form

$$x(t) = \sum_{k=-\infty}^{\infty} g(t - kT) \quad (17.41)$$

with the Fourier transform of  $g(t)$ . In the process, we will get *Poisson's sum formula*, a well known result. Any periodic signal with period  $T$  can be written in summation form (17.41).

Because of the periodicity of  $x(t)$ , we can express it using a Fourier series

$$x(t) = \sum_{m=-\infty}^{\infty} X_m e^{j2\pi m t/T} \quad (17.42)$$

where the Fourier coefficients are

$$X_m = \frac{1}{T} \int_{-T/2}^{T/2} \left[ \sum_{k=-\infty}^{\infty} g(t - kT) \right] e^{-j2\pi m t/T} dt. \quad (17.43)$$

#### Exercise 17-7.

Show that (17.43) reduces to

$$X_m = \frac{1}{T} \int_{-\infty}^{\infty} g(\tau) e^{-j2\pi m \tau/T} d\tau. \quad (17.44)$$

□

Except for the  $1/T$  factor, this is the Fourier transform of  $g(t)$  evaluated at  $\omega = 2\pi m/T$

$$X_m = \frac{1}{T} G(j2\pi m/T). \quad (17.45)$$

Thus, the Fourier coefficients of the periodic signal in summation form are simply samples of the Fourier transform of the function  $g(t)$ . Putting (17.45) together with (17.42) we get Poisson's sum formula

$$\sum_{k=-\infty}^{\infty} g(t - kT) = \frac{1}{T} \sum_{m=-\infty}^{\infty} G(j\frac{2\pi}{T}m) e^{j2\pi m t/T}. \quad (17.46)$$

## APPENDIX 17-B DISCRETE-TIME DERIVATIVE

Consider a continuous-time signal  $Q(t)$  for which we have available only Nyquist-rate samples

$$Q_k(\tau) = Q(kT + \tau) \quad (17.47)$$

where  $T$  is the sampling interval and  $\tau$  is the sampling phase. If the sampling phase is varying with time, we must assume it is varying slowly enough that we can consider it essentially constant. We wish to compute the derivative of  $Q_k(\tau)$  with respect to  $\tau$  for all  $k$ . From the interpolation formula (2.19) we can write

$$Q(t) = \sum_{m=-\infty}^{\infty} Q_m(\tau) \frac{\sin \frac{\pi}{T}(t - mT - \tau)}{\frac{\pi}{T}(t - mT - \tau)}. \quad (17.48)$$

The derivative of  $Q_k(\tau)$  with respect to the timing phase  $\tau$  is the same as the derivative of  $Q(t)$  sampled at the same sampling instants,

$$\begin{aligned} \frac{\partial Q_k(\tau)}{\partial \tau} &= \left[ \frac{\partial Q(t)}{\partial t} \right]_{t=kT+\tau} \\ &= \sum_{m=-\infty}^{\infty} Q_m(\tau) \left[ \frac{\cos \frac{\pi}{T}(t-kT-\tau)}{\frac{\pi}{T}(t-kT-\tau)} - \frac{\sin \frac{\pi}{T}(t-kT-\tau)}{\frac{\pi}{T}(t-kT-\tau)^2} \right]_{t=kT+\tau} \quad (17.49) \\ &= \sum_{m, m \neq k} z_m(\tau) \left[ \frac{(-1)^{k-m}}{(k-m)T} \right]. \end{aligned}$$

The last equality follows from putting the quantity in brackets over a common denominator, and evaluating everywhere except at  $m = k$ . At that point, use of L'Hospital's rule shows that the value is zero. The sum in (17.49) can be rewritten as a convolution

$$\frac{\partial Q_k(\tau)}{\partial \tau} = Q_k(\tau) * d_k \quad (17.50)$$

where

$$d_k = \begin{cases} 0, & k = 0 \\ \frac{(-1)^k}{kT}, & k \neq 0 \end{cases} \quad (17.51)$$

Hence the derivative of the sampled signal with respect to the timing phase can be obtained by filtering the sampled signal with the filter whose impulse response is given by (17.51).

A filter with impulse response given by (17.51) would be difficult to implement. In practice, the impulse response can be truncated, and the filter can be implemented as an FIR filter. Reducing it to three taps, we get the approximation

$$d_k \approx (\delta_{k+1} - \delta_{k-1})/T, \quad \frac{\partial Q_k(\tau)}{\partial \tau} \approx (Q_{k+1}(\tau) - Q_{k-1}(\tau))/T. \quad (17.52)$$

Note, however, that the impulse response in (17.51) decays only linearly with increasing  $|k|$ , so any truncated FIR filter will have substantial error.

## PROBLEMS

- 17-1. Consider a signal  $Q(t)$  bandlimited to  $|\omega| < \pi/T$  and samples of the signal  $Q_k(\tau)$  given by

$$Q_k(\tau) = Q(kT + \tau)$$

where the sampling phase  $\tau$  is arbitrary. Give the impulse response of a filter whose input is  $Q_k(\tau)$  and output is  $Q_k(\alpha)$ , for any arbitrary  $\alpha$  different from  $\tau$ . This suggests that given Nyquist-rate samples of a received signal, an equalizer can compensate for any timing phase. If the equalizer is adaptive, it will automatically compensate for an erroneous timing phase. Most adaptive filters, however, are implemented as FIR filters. Can this filter be exactly implemented or approximated as an FIR filter?

- 17-2. Suppose a received real-valued baseband PAM signal has 100% excess bandwidth raised cosine Nyquist pulses given in (6.24) (with  $\alpha = 1$ ). Find the Fourier series coefficients of  $E[R^2(t)]$ . Would square law timing recovery work well with this signal?
- 17-3. Show that a baseband real-valued PAM signal with a 0% excess bandwidth raised cosine Nyquist pulse has no timing tone for the squarer timing recovery circuit of Figure 17-6. Note that although it has a timing tone for a fourth-power circuit, it has a zero-width eye (shown in Problem 6-5) and hence cannot be used anyway.
- 17-4. Consider the WSS random process  $R(t + \Theta)$  where  $R(t)$  is defined by (17.2) and  $\Theta$  is a uniformly distributed random variable over the range 0 to  $T$ . Assume  $A_k$  is white,  $R_A(m) = a\delta_m$  (which implies zero mean), and independent of  $\Theta$ . Show that  $R(t + \Theta)$  cannot have any spectral lines not present in  $p(t)$ . In other words, in order to get a spectral line at the symbol frequency,  $p(t)$  would have to be periodic with period  $T$ .
- 17-5. Consider a 600 baud signal with 100% excess bandwidth. Design a prefilter for square law timing recovery. Will the prefilter improve performance?
- 17-6. Consider the inductive timing recovery of Figure 17-3. Write the output of the VCO

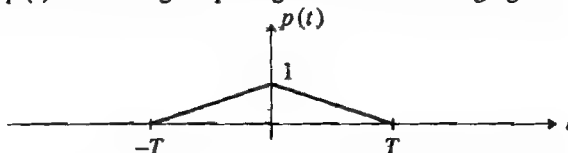
$$v(t) = \cos(\omega_s t + \phi(t)). \quad (17.53)$$

Samples of the input signal are taken at times  $kT + \tau_k$ , which correspond to the zero crossings of  $v(t)$ . Consider only the zero crossings from positive to negative, so for small enough  $\epsilon > 0$ ,

$$\begin{aligned} v(kT + \tau_k) &= 0 \\ v(kT + \tau_k - \epsilon) &< 0 \\ v(kT + \tau_k + \epsilon) &> 0. \end{aligned} \quad (17.54)$$

Find an equation relating  $\tau_k$  and  $\phi(t)$ .

- 17-7. Suppose that the input to the slicer is given by (17.32). Assume that  $A_k$  is real-valued, zero mean, white, and independent of  $N_k$ , which also has zero mean. For an implicit timing recovery technique using the update given by (17.27), (17.35), and (17.37),
- Show that the timing function (17.29) is given by (17.33).
  - Suppose that  $p(t)$  is the triangular pulse given in the following figure:



Sketch the timing function.

- 17-8. Suppose again that the input to the slicer is given by (17.32), where  $A_k$  is real-valued, zero mean, white, and independent of  $N_k$ . For an implicit timing recovery technique using the update given by (17.27), (17.35), and

$$\mathbf{X}_k' = \mathbf{A}_{k-1} . \quad (17.55)$$

- (a) Find the timing function (17.29).
- (b) Suppose that  $p(t)$  is the same triangular pulse used in Problem 17-7. Sketch the timing function.
- (c) Will this timing update work with the triangular pulse? Will it work with raised cosine pulses?
- (d) Repeat parts (a) through (c) for

$$\mathbf{X}_k' = \mathbf{A}_{k+1} . \quad (17.56)$$

- 17-9. Explain the  $N$  dependence of the mean-square jitter in Example 17-13 by examining the bandwidth over which the total jitter transfer function is approximately  $N$ . In particular, show that this bandwidth is proportional to  $1/N$ , and thus, the jitter power is proportional to  $N^2 \cdot (1/N) = N$ . Thus, the jitter does not accumulate at nearly the same rate as for an ideal lowpass filter of Example 17-12. (Hint: Do a Taylor series expansion of  $H^N(j\omega)$ , retain only first- and second-order terms.)

## REFERENCES

1. P. Kabal and S. Pasupathy, "Partial-Response Signaling," *IEEE Trans. on Communications* COM-23(9)(Sep. 1975).
2. W. R. Bennett, "Statistics of Regenerative Data Transmission," *BSTJ* 37 pp. 1501-1542 (Nov. 1958).
3. L. E. Franks, "Synchronization Subsystems: Analysis and Design," in *Digital Communications: Satellite/Earth Station Engineering*, Prentice-Hall Inc. (1981).
4. N. A. D'Andrea and U. Mengali, "A Simulations Study of Clock Recovery in QPSK and 9QPSK Systems," *IEEE Trans. on Communications* COM-33(10)(Oct. 1985).
5. H. Kobayashi, "Simultaneous Adaptive Estimation and Decision Algorithm for Carrier Modulated Data Transmission Systems," *IEEE Trans. on Communications Technology* COM-19 pp. 268-280 (June 1971).
6. S. U. H. Qureshi, "Timing Recovery for Equalized Partial-Response Systems," *IEEE Trans. on Communications*, (Dec. 1976).
7. H. Sari, L. Desperben, and S. Moridi, "Optimum Timing Recovery for Digital Equalizers," *Globecom '85 Proceedings*, (1985).
8. R. D. Gitlin and J. Salz, "Timing Recovery in PAM Systems," *BSTJ* 50(5) p. 1645 (May and June 1971).
9. B. R. Saltzberg, "Timing Recovery for Synchronous Binary Data Transmission," *BSTJ*, pp. 593-622 (March 1967).
10. K. H. Mueller and M. Muller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Trans. on Communications* COM-24 pp. 516-531 (May 1976).
11. C. J. Byrne, B. J. Karafin, and D. B. Robinson, "Systematic Jitter in a Chain of Digital Repeaters," *Bell Sys. Tech. J.* 42 p. 2679 (Nov. 1963).
12. Bell Laboratories Members of Technical Staff, *Transmission Systems for Communications*, Western Electric Co., Winston-Salem N.C. (1970).
13. R. D. Gitlin and J. F. Hayes, "Timing Recovery and Scramblers in Data Transmission," *BSTJ* 54(3)(March 1975).

14. M. R. Aaron, "PCM Transmission in the Exchange Plant," *BSTJ* **41** pp. 99-141 (Jan. 1962).
15. G. Ungerboeck, "Fractional Tap-Spacing and Consequences for Clock Recovery in Data Modems," *IEEE Trans. on Communications*, (Aug. 1976).
16. R. D. Gitlin and S. B. Weinstein, "Fractionally-Spaced Equalization: An Improved Digital Transversal Equalizer," *BSTJ* **60**(2)(Feb. 1981).
17. O. Agazzi, C.-P. J. Tzeng, D. G. Messerschmitt, and D. A. Hodges, "Timing Recovery in Digital Subscriber Loops," *IEEE Trans. on Communications* **COM-33**(6)(June 1985).
18. A. Jennings and B. R. Clarke, "Data-Sequence Selective Timing Recovery for PAM Systems," *IEEE Trans. on Communications* **COM-33**(7)(July 1985).
19. D. D. Falconer, "Timing Jitter Effects on Digital Subscriber Loop Echo Cancellers: Part I - Analysis of the Effect," *IEEE Trans. on Communications* **COM-33**(8)(Aug. 1985).
20. D. D. Falconer, "Timing Jitter Effects on Digital Subscriber Loop Echo Cancellers: Part II - Considerations for Squaring Loop Timing Recovery," *IEEE Trans. on Communications* **COM-33**(8)(Aug. 1985).
21. D. G. Messerschmitt, "Asynchronous and Timing-Jitter Insensitive Data Echo Cancellation," *IEEE Trans. on Communications* **COM-34**(12) p. 1209 (Dec. 1986).
22. C. M. Monti and G. L. Pierobon, "Block Codes for Linear Timing Recovery in Data Transmission Systems," *IEEE Trans. on Communications* **COM-33**(6)(June 1985).
23. E. D. Sunde, "Self-Timing Regenerative Repeaters," *BSTJ* **36** pp. 891-937 (July 1957).
24. A. Haoui, H.-H. Lu, and D. Hedberg, "An All-Digital Timing Recovery Scheme for Voiceband Data Modems," *Proceedings of ICASSP*, (1987).
25. D. L. Lyon, "Timing Recovery in Synchronous Equalized Data Communication," *IEEE Trans. on Communications* **COM-23**(2)(Feb. 1975).

# 18

---

## MULTIPLE ACCESS ALTERNATIVES

---

Thus far in this book we have discussed the signal processing necessary for digital communications from one point to another over a communications medium. The remainder of the book will begin to address the realization of a *digital communications network* in which many users simultaneously communicate with one another. One of the key issues that must be resolved in moving from a single digital communication system to a network is how we provide access to a single transmission medium for two or more (typically many more) users. In moving from a single transmitter and receiver to the sharing of media by multiple users, we must address two primary issues. First, how do we resolve the *contention* that is inherent in sharing a single resource. This issue is discussed in this chapter. Secondly, how do we *synchronize* all the users of the network as an aid to resolving the contention. In Chapter 19 we describe one particular multiple access technique based on signal processing called *echo cancellation* that is specifically used for *full-duplex* transmission.

Important practical applications of multiple access techniques include:

- *Full-duplex data transmission on a single medium.* Here we want the two directions of transmission to share a single transmission medium such as a wire pair or voiceband channel. An important application of this is data transmission on the subscriber loop, between telephone central office and customer premises, where only a single wire pair is available for both directions of transmission.
- *Multiple channels sharing a common high-speed transmission link.* By building very high speed transmission links using, for example, coaxial cable or optical



fiber, and then sharing that link over many users, economies of scale are realized. The cost of the transmission link when divided by the number of users can be very much lower than the cost of an equivalent lower speed link. The process of sharing many channels on a single high speed link is called *multiplexing*. A network, which provides a communications capability between any pair of users on demand, can be constructed from communications links, multiplexers, and *switches*. The latter provide the rearrangement of the connections through the network necessary to provide this service on demand.

- Many users can share a common transmission medium in such a way that when one user broadcasts to the others, with only the user for which the communication is intended paying attention. By this means, it is possible to provide access between any user and any other user without a switch. This is commonly known as *multiple access*, although we use this term more generally in this part of the book to describe any situation where two or more users share a common transmission medium. Particularly for small numbers of users and within limited geographical areas, multiple access to a single medium can be more economical than using switches. This type of multiple access is often exploited in *local-area networks* within a customer premises, and in *satellite networks* in the context of a wide geographical area.

All these multiple access techniques require that the messages corresponding to different users be separated in some fashion so that they do not interfere with one another. This is usually accomplished by making the messages *orthogonal* to one another in signal space. We can then separate out the different signals using some form of matched filtering or its equivalent, which because of the orthogonality of the signals will respond to only a single signal.

Orthogonality of two or more signals can be accomplished in several ways.

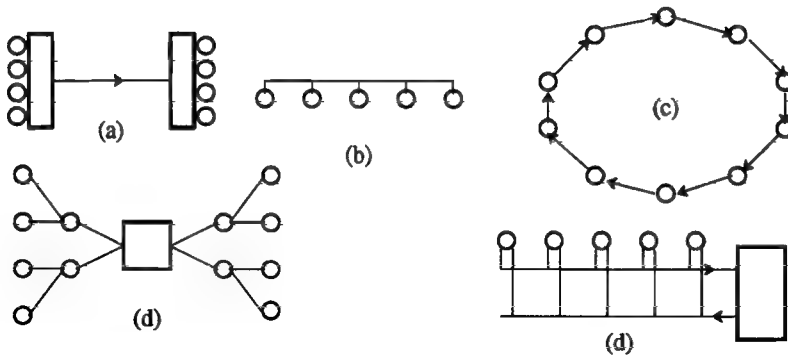
- The messages can be separated in *time*, insuring that the different users transmit at different times.
- The messages can be separated in *frequency*, insuring that the different users use different frequency bands.
- The messages can be transmitted at the same time and at the same frequency, but made orthogonal by some other means. Usually this is done by *code division*, in which the users transmit signals which are guaranteed to be orthogonal through the use of specially designed codes.

There are also cases where the signals are not separated by orthogonality. In the particular case of full duplex transmission on a common medium, the signal which is interfering with the received data stream is the transmit data stream generated by the same user. We can use the fact that the interfering signal is known, and use *echo cancellation* to separate the two directions, even though they are not orthogonal (although we do use the fact that they are uncorrelated). Echo cancellation is a very special technique that is discussed in detail in Chapter 19. In this chapter we cover the first three multiple access techniques, separation by time, frequency, and code division. We also describe the cellular concept, which allows spatial reuse of frequencies for serving large numbers of users.

## 18.1. MEDIUM TOPOLOGY FOR MULTIPLE ACCESS

In any discussion of multiple access techniques, it is appropriate to begin by pointing out the importance of the *topologies* of the medium. This term refers to the geometrical configuration of the medium which is being shared by two or more users. Each of the common media have preferred topological configurations, and each topology suggests appropriate techniques for providing multiple access. Multiple access media are most common over geographically limited areas, such as the local-area and metropolitan-area networks, but can also span large geographical areas when radio and satellite media are used. In this section we discuss briefly the most common topologies, and in the remainder of the chapter describe multiple access techniques in the context of these topologies.

Some representative topologies are shown in Figure 18-1. In a the simplest and most common situation is shown, where we have a single uni-directional communications *link* that we wish to share over two or more users. To do so we implement a *multiplexer* and *demultiplexer*, represented by the boxes, which accept information from the users and transmit it over the link. This topology inherently has *contention*, in that two or more users may wish to use the medium simultaneously. The multiplexer, since it has a form of central control, can easily avoid *collisions* (two or more users transmitting simultaneously) since it controls what is transmitted. In Figure 18-1b we show the *bus*, in which two or more users are connected in parallel to a common medium, with the result that every transmission by any user is received by every other user. The most common medium for a bus is coaxial cable or wire-pair, where



**Figure 18-1.** Common topologies for multiple access. A line corresponds to a communication link, one-way if it includes an arrow, a box is a central controller that controls access to the medium, and a circle represents a user. (a) A link, where multiple users share a common one-way communication channel using a multiplexer and demultiplexer. (b) A bus, where every user receives the signal from every other user. (c) A ring, in which each user talks to one neighboring user. (d) A tree, in which users communicate through a central hub. (e) A bus with centralized control.

the users are simply connected in parallel. A set of radio transmitters and receivers with non-directional antennas are also connected, in effect, by a bus, as are a set of transmitters and receivers communicating through a satellite transponder. In the later cases this is known as a *broadcast medium*. The bus has no central control, which makes it more difficult to avoid collisions. The *ring* is shown in Figure 18-1c, where the users are connected to their nearest neighbor in a circle. In this case, we cannot allow broadcast, since the information would circulate indefinitely. Therefore, we must have *active nodes* in this topology, which means that all communications pass through them rather than just by them. This allows users to remove communications as well as initiate them (typically they remove their own transmissions so that they will not circulate forever). Rings are the topology of choice for local area networks based on fiber optics media, since they require only point-to-point connections and the bus topology is difficult to realize using optical components. The *tree* is shown in Figure 18-1d, in which all users are connected to one another through a central controller. Finally, the *bus with central control* is shown in Figure 18-1e. This topology is very similar to the tree in that all users communicate through a central controller, and requires two uni-directional busses.

A special application of multiple access is the sharing of a single medium for digital transmission in both directions. This is called *full-duplex* transmission, and is pictured in Figure 18-2. We have two transmitters, one on the left, and one on the right, and two receivers. The goal is to have the transmitters on each end transmit to the receivers on the opposite end. Unfortunately, most media have the characteristic that each receiver must contend with interference from the local transmitter in addition to the signal from the remote transmitter. In fact, it will often be the case that the local transmitter interference is much larger than the remote transmitter signal. The receiver must find some way to separate the local and remote transmitter signals.

When data signals are transmitted through the network, they encounter echos at points of four-wire to two-wire conversion. In *half-duplex* data transmission (in only one direction), echos present no problem since there is no receiver on the transmitting end to be affected by the echo. In full-duplex transmission, where the data signals are transmitted in both directions simultaneously, echos from the data signal transmitted in one direction interfere with the data signal in the opposite direction as illustrated in Figure 5-39.

Most digital transmission is half-duplex. For example, the high speed trunk digital transmission systems separate the two directions of transmission on physically different wire, coaxial, or fiber optic media. The two directions therefore do not

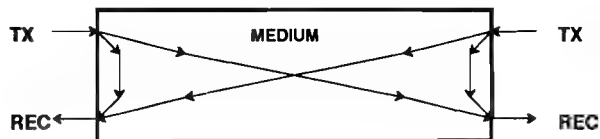


Figure 18-2. Illustration of full-duplex transmission.

interfere, except perhaps through crosstalk resulting from inductive or capacitive coupling. However, full-duplex data transmission over a common media has arisen in two important applications. In both these applications, the need for a common media arises because the network often only provides a two-wire connection to each customer premise (this is because of the high cost of copper wire, and the large percentage of the telephone network investment in this facility).

#### Example 18-1.

The first application shown in Example 18-1 is digital transmission on the *subscriber loop*, in which the basic voice service and enhanced data services are provided through the two-wire subscriber loop. Total bit rates for this application that have been proposed are 80 and 144 kb/s in each direction, where the latter rate includes provision for two voice/data channels at 64 kb/s each plus a data channel at 16 kb/s, and the first alternative allows only a single 64 kb/s voice/data channel. This digital subscriber loop capability is an important element of the emerging integrated services digital network (ISDN), in which integrated voice and data services will be provided to the customer over a common facility [1]. Voice transmission requires a *codec* (coder-decoder) and anti-aliasing and reconstruction filters to perform the analog-to-digital and digital-to-analog conversion on the customer premises, together with a *transceiver* (transmitter-receiver) for transmitting the full-duplex data stream over the two-wire subscriber loop. Any data signals to be accommodated are simply connected directly to the transceiver. The central office end of the loop has another full-duplex transceiver, with connections to the digital central office switch for voice or circuit-switched data transmission, and to data networks for packet switched data transport capability. □

#### Example 18-2.

The second application for full-duplex data transmission is in *voiceband data transmission*, already illustrated in Figure 5-39. The basic customer interface to the network is usually the same two-wire subscriber loop. In this case the transmission link is usually **more complicated** due to the possible presence of four-wire trunk facilities in the middle of the connection. The situation can be even more complicated by the presence of two-wire toll switches, allowing intermediate four-two-four wire conversions internal to the network. □

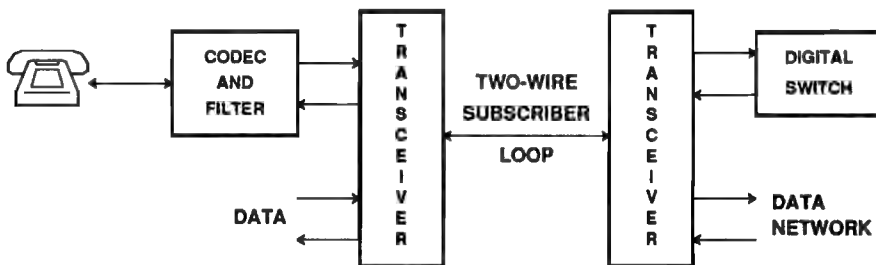


Figure 18-3. A digital subscriber loop transceiver for full-duplex digital transmission.

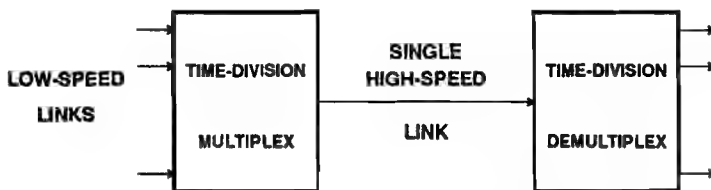
The two applications differ substantially in the types of problems which must be overcome. For the digital subscriber loop, the transmission medium is fairly ideal, consisting of wire pairs with a wide bandwidth capability. The biggest complication is the higher bit rate and the presence in some countries of bridged taps — open-circuited wire pairs bridged onto the main line. The voiceband data modem, while requiring a lower speed of transmission, encounters many more impairments. In addition to the severe bandlimiting when carrier facilities are used, there are problems with noise, nonlinearities, and sometimes even frequency offset. Another difference is that the subscriber loop can use baseband transmission, while the voiceband data set always uses passband transmission.

## 18.2. MULTIPLE ACCESS BY TIME DIVISION

By far the most common method of separating channels or users on a common digital communications medium is by ensuring that they transmit at different times. This is known as *multiple access by time-division*. This technique has many variations, the most common of which are described in this section. In all these variations, some method is used to avoid *collisions*, or two or more users transmitting simultaneously. Collision avoidance in link access is somewhat easier than in the other topologies, and therefore we discuss link access separately.

### 18.2.1. Point-to-Point Link Access

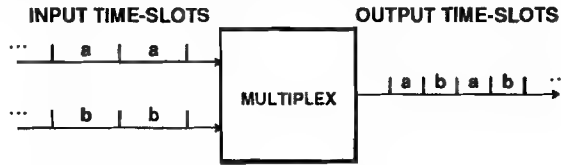
It is often desired to divide a high-speed bit stream over a point-to-point communications link into a set of lower-rate bit streams, each with a fixed and predefined bit rate. Where this is desired, it is appropriate to use a technique called *time-division multiplexing (TDM)*. The bit streams to be multiplexed are called *tributary streams*. Where these tributary bit-streams are provided directly to a user, that is they do not themselves consist of tributary streams, then they are called *circuits* or *connections*. We *interleave* these tributary streams to obtain a higher rate bit stream. The purpose of the multiplex, shown functionally in Figure 18-4, is to take advantage of the economies of scale of a high-speed transmission system.



**Figure 18-4.** A time-division multiplex, which interleaves a number of lower-speed tributaries on a single higher-speed link.

**Example 18-3.**

A simple multiplexing function for two tributary streams is shown below:



Each tributary stream is divided continuously into groups of bits, known as *time-slots*, and then these time-slots are interleaved to form the output bit stream. Each slot on the output bit stream occupies half the time of an input slot since the bit rate is twice as great. □

In practice any number of tributary streams can be multiplexed, and a defined time-slot can have any number of bits. However, two cases are particularly common: a time-slot equal to one bit, known as *bit-interleaving*, and a time-slot equal to eight bits, which is known as *octet-interleaving*. In multiplexing, an *octet* is a common term for eight bits. Organization around octets is common because voice PCM systems commonly use eight bits per sample quantization, and because data communications systems typically transfer eight-bit groupings of bits, known in the computer world as *bytes*. In both cases it is necessary to maintain *octet integrity* at the destination, meaning that the bit stream is delimited into the same eight-bit boundaries defined at the origin. This octet integrity is assured by using octet-interleaving, although this is not the only means.

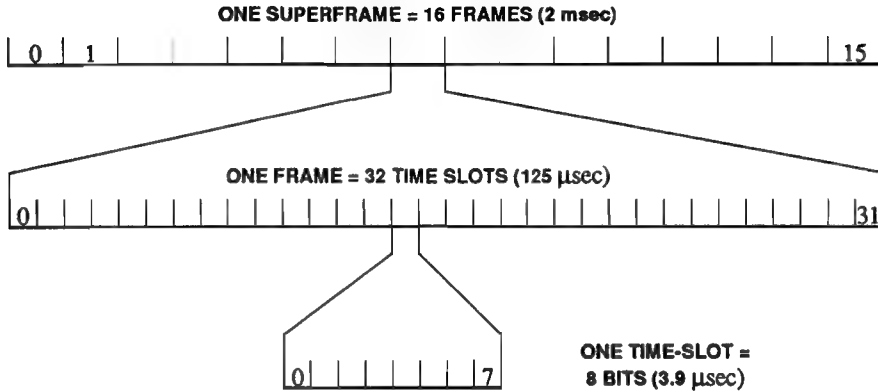
On the high-speed output bit stream, the collection of bits corresponding to precisely one time-slot from each tributary stream is known as a *frame*. In Example 18-3 one frame corresponds to time-slots *a* and *b*. At the demultiplex, all we have is a bit-stream originating at the multiplex. In order to realize the demultiplexing function, the boundaries of the time-slots must be known. Furthermore, to ensure that the correspondence between input and output tributary streams is maintained, demultiplexing requires knowledge of the beginning of the frame. For this purpose, the multiplex typically inserts additional bits into the frame known as *framing bits*.

**Example 18-4.**

In a time-division multiplex,  $N$  tributary streams are multiplexed with  $M$ -bit time-slots into a single output bit stream. The number of bits in the output frame is  $N \cdot M$  plus any added framing bits. □

The framing bits follow a deterministic pattern which can be recognized at the demultiplex as distinct from the information bits. Once the demultiplex has located these bits, through a process known as *framing recovery*, it has a reference point that enables it to locate the beginning of the frame.

Since a multiplex cannot store an unbounded number of bits, we must ensure that the minimum bit rate of the output high-speed stream is greater than or equal to the sum of the maximum bit rates rates of the tributary streams plus the rate required for framing and any other overhead bits.



**Figure 18-5.** The frame structure for the CCITT G.732 30-channel PCM system.

**Example 18-5.**

The CCITT 30-channel system (recommendation G.732 [2]) is widely used in Europe and multiplexes 30 tributary streams, each at 64 kb/s, appropriate for a voiceband channel, into a single 2048 kb/s bit stream. Note that  $30 \cdot 64 = 1920$ , so that 128 kb/s is used for overhead functions such as framing. The organization of the frame is shown in Figure 18-5. Each frame is divided into 32 eight-bit time slots, 30 of them taken from the tributary streams, and the remaining two used for overhead. Thus, in this case as in the case of most lower-speed multiplexes, octet-interleaving is used. The time for one frame corresponds to an octet on each tributary stream, or 1.25 μsec. There is also defined a *superframe* or *multiframe* of 16 frames, which is used to transmit and frame *on-off hook information* for each of the 30 tributary voiceband channels. This on-off hook information, transmitted in frames 0 and 16, is used to communicate between switching machines during call setup and take-down. Time-slot 0 always contains the octet "x0011011" and "x10xxxxx" in alternate frames, indicating the beginning of the frame, and time-slot 16 contains "0000x0xx" in frame 0 of the superframe indicating the beginning of the superframe ("x" indicates bits not assigned, which can be used for other purposes). Time-slot 16 in the remaining frames of the superframe contains the aforementioned signaling information. □

**Example 18-6.**

The CCITT 24-channel system used in North America (CCITT G.733 [2]) has a frame consisting of 193 bits, including 24 eight-bit time-slots for the tributary 64 kb/s channels and one framing/superframing bit. In a superframe of 12 frames, the framing bit contains the pattern "101010", the framing pattern, interleaved with the pattern "001110", the super-frame pattern. The bit rate is  $193 \cdot 8 = 1544$  kb/s. □

**Example 18-7.**

The M12 multiplex used in the North American network multiplexes four tributary bit streams at 1544 kb/s (often the G.733 signal of Example 18-6) into a 1176-bit super-frame shown in Table 18-1 using bit interleaving. Each line of the table represents one

$M_0$	48I	$C_1$	48I	$F_0$	48I	$C_1$	48I	$C_1$	48I	$F_1$	48I
$M_1$	48I	$C_2$	48I	$F_0$	48I	$C_2$	48I	$C_2$	48I	$F_1$	48I
$M_1$	48I	$C_3$	48I	$F_0$	48I	$C_3$	48I	$C_3$	48I	$F_1$	48I
$M_1$	48I	$C_4$	48I	$F_0$	48I	$C_4$	48I	$C_4$	48I	$F_1$	48I

**Table 18-1.** Superframe structure of the M12 multiplex. The  $F$  bits are the framing bits,  $M$  are the superframing bits, and  $C$  are stuffing control bits. 48I means 48 information bits, 12 bit-interleaved bits from each of four tributary streams.

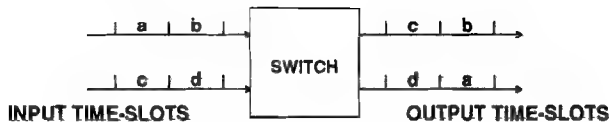
frame as defined by the  $F_0 F_1 \cdots$  pattern, where  $F_0 = 0$  and  $F_1 = 1$ . Similarly, the four-frame superframe is defined by the  $M_0 M_1 M_1 M_1 \cdots$  superframe pattern.  $\square$

## Digital Circuit Switching

TDM multiplexing provides a basic capability for the network to provide fixed bit-rate bit streams between users. These bit streams, which can be used to provide services such as a voiceband channel or video channel, are called *circuits* or *connections* through the network. This terminology comes from the early days of the telephone network, when voiceband channels were provided by a continuous metallic connection. In order for the network to provide the precise set of circuits requested by the users at any time, it is necessary to provide a set of *digital circuit switches*. The function of a digital circuit switch, and particularly the transmission interfaces, are similar to that of a TDM multiplex, so they will be described briefly here. The switch differs from the multiplex in that it typically has the same number of output bit streams as inputs. Typically each stream is itself composed of time-division multiplexed lower-speed streams, so that it has a defined framing structure with time-slots corresponding to each tributary stream. Further, each of these time-slots typically corresponds to a circuit, or bit stream provided directly to users, since the purpose of the digital circuit switch is to connect different users together. The specific purpose of the switch is to perform an arbitrary permutation of the input circuits (time-slots) appearing on the output.

### Example 18-8.

A simple example is shown below:



Each of the input time-slots represents a circuit, where there are two circuits corresponding to each TDM input and output in this example. Thus, in total there are four input and four output circuits, and the purpose of the switch is to perform an arbitrary permutation of the input circuits as they appear at the output. (If this switch were used for voiceband channels, each tributary would be 64 kb/s). In the figure, input circuit  $a$  replaces  $d$  at the output,  $c$  replaces  $a$ , etc. To perform its function, the switch must be able to transfer the bits corresponding to one time-slot on one of the input TDM streams to a time-slot on a



different output TDM stream (for example **a** and **c** in the figure), which is known as *space-division switching*. Also, it must be able to transfer the bits from one time-slots to another (for example **a** and **d** in the figure), which is known as *time-division switching*. □

Of course, most practical switches are much larger than this example.

#### Example 18-9.

The No. 4 ESS is a large toll digital switching machine used in the North American network. It has 5120 input bit streams, each of which is the G.733 signal of Example 18-6 containing 24 tributary voiceband channels, for a total of 122,880 tributary 64 kb/s bit streams (the total input and output bit rate is 7.86 Gb/s, or actually double this because each channel is bi-directional). This switch is not capable of providing all possible permutations of input-output connections, but reasonable traffic demands can be served with high probability. □

*Circuit switching* enables the network to provide a connection between two users for the duration of their need. When the circuit is no longer needed, the switch terminates the connection and uses the associated time-slots on the transmission facility to provide another connection. In this fashion, we avoid provisioning transmission capacity for every possible connection in the network, but rather provide only sufficient capacity for those connections likely to be required at any give time. In Section 18.2.3 we will see an alternative model for providing a connection between two users, called packet switching.

### 18.2.2. Time-Division Multiple Access

Thus far this section has addressed the use of time-division techniques for access control to a point-to-point link and to a full-duplex channel. Time-division using the circuit switching approach can also be applied to other multiple access topologies, such as the bus or the ring. The appropriate technique is highly dependent on the topology. For example, on the ring topology it is straightforward to define a scheme for time-division multiplexing.

#### Exercise 18-1.

Describe a method for forming a frame and fixed time-slot structure on a ring topology, and the way in which a circuit could be formed. This approach is called a *slotted ring*. □

On the other hand, TDM is difficult to apply to the bus topology. The reason is simply that TDM requires a fixed frame known to all nodes of the network, but in a bus, particularly a geographically large broadcast network such as a satellite network, the propagation delays on the medium will typically be large relative to one bit-time. It is still common to apply TDM techniques in this situation, but they must be modified to account for the significant propagation delays between users. This modification leads to an approach known as *time-division multiple access (TDMA)*. TDMA is applicable to any bus or broadcast topology, where there is a set of transmitters and a set of receivers, all of which hear each transmission. It has been extensively applied to satellite networks in particular [3,4].

TDMA requires a centralized control node, a feature that can be avoided using the random access techniques discussed in Section 18.2.4. The primary functions of the control node are to transmit a periodic *reference burst*, akin to the added framing bits in TDM, that defines a frame and forces a measure of synchronization of all the other nodes. The frame so-defined is divided into time-slots, as in TDM, and each node is assigned a unique time-slot in which to transmit its information. The resulting frame structure is illustrated in Figure 18-6, including one frame plus the succeeding reference burst. Each node, of which there are  $N$ , transmits a *traffic burst* within its assigned time-slot. Thus far, the approach is similar to TDM, but there are significant differences. First, since there are significant delays between nodes, each node receives the reference burst with a different phase, and thus its traffic burst is transmitted with a correspondingly different phase within the time slot. There is therefore a need for *guard times* to take account of this uncertainty. Each time-slot is therefore longer than the period needed for the actual traffic burst, thereby avoiding the overlap of traffic bursts even in the face of these propagation delays. Second, since each traffic burst is transmitted independently with an uncertain phase relative to the reference burst, there is the need for a *preamble* at the beginning of each traffic burst to allow the receiver to acquire timing and carrier phase. Finally, there must be a centralized control mechanism to assign time-slots and communicate those assignments to the nodes. Time-slots can be *pre-assigned*, implying that changes are infrequent and only as a result of rearrangements, or *demand-assigned*, meaning that frequent reassignments are made to match the ebb and flow of traffic demands.

The reference burst is composed of three parts. A deterministic *carrier and bit-timing* sequence of approximately 30 to 300 symbols enables each node to do accurate carrier recovery and timing recovery for detection of the subsequent information bits. This is followed by a *unique word* with good autocorrelation properties, enabling each node to establish an accurate time-reference within the reference burst. The unique word is entirely analogous to the added framing bits in a TDM frame. Finally, there is the *control and delay channel*, which is a set of information bits used for control of the nodes. It enables the central control to assign time-slots, and can even be used to control the phase of a node's traffic burst within a time-slot, thereby reducing the size of the guard time. The traffic burst preamble has a very similar structure. The control

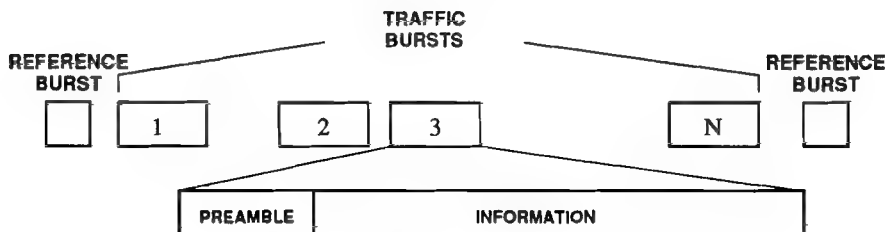


Figure 18-6. The frame structure of a TDMA system.

algorithms can get fairly complicated, and the reader is referred to [3] for a more detailed discussion.

Early satellite systems utilized multiple access by FDM, to be described later, but the current trend is to use TDMA.

**Example 18-10.**

The INTELSAT system uses TDMA for high-volume international traffic. The basic bit rate is 120.832 Mb/s using four-phase PSK so that the baud rate is 60.416 MHz. The basic frame is 2 ms in length, this being 16 times the frame period of both of the standard primary multiplex standards in the world (Example 18-5 and Example 18-6). Each time-slot is assigned to one of these primary bit streams. For a G.732 bit stream (Example 18-5), the traffic burst will contain  $2 \text{ ms} \cdot 2.048 \text{ kb/s} = 4096$  information bits. At a bit rate of 120.832 Mb/s, the information portion of the traffic burst consumes 33.9  $\mu\text{sec}$ . Forgetting the reference burst, guard times, and traffic burst preambles, the frame has room for 59 of these G.732 bit streams.  $\square$

### 18.2.3. Packetizing

The circuit switching approach described in Section 18.1.1 allocated a fixed bit rate, corresponding to a reserved time-slot, to each of the multiple users of a given transmission system. This approach is simple to implement, but also cannot provide a time-varying bandwidth or a bandwidth on demand. It is also inflexible in that a fixed maximum number of users can be accommodated on any given transmission system. There are many examples of services that inherently require a time-varying bit rate, and in this case using circuit switching we must provide a circuit commensurate with the *maximum* bit rate requirements. Circuit switching is therefore somewhat inefficient for these services.

**Example 18-11.**

In conversational speech, normally only one direction of the conversation is active at any given time. During the resulting silence intervals, a smaller or even zero transmission capacity is required. Circuit switching is therefore no better than 50% efficient.  $\square$

**Example 18-12.**

In video transmission, the image can be reconstructed with high accuracy from past images during periods of limited motion. Therefore, for a given fidelity higher bit rates are inherently required during periods of high motion than during periods of reduced motion.  $\square$

**Example 18-13.**

In interactive data transmission, transmission capacity from a user typing at a keyboard is sporadic, depending on when keys are pressed. For this case, circuit switching is grossly inefficient.  $\square$

**Example 18-14.**

An alarm system connected to an alarm service bureau requires transmission capacity only infrequently to transmit "all is well" messages and even more infrequent "panic" messages.

□

An additional problem with circuit switching is that mixing different services with different bit rate requirements, even when their bit rates are constant with time, becomes administratively complicated.

*Packet switching* or *store-and-forward switching* provides these capabilities lacking in circuit switching. It allows us to mix bit streams from different users that vary in bit rate, and dynamically allocates the available bandwidth among these users. In this subsection we will describe the simple technique of packetizing as applied to multiplexing multiple users onto a single communications link, and in the following subsection we discuss multiple access using packetizing.

The basic idea of packetizing a bit stream as a flexible approach for multiplexing a number of such streams together is very simple. The information bits from each user are divided into groups, called *information packets* or just *packets*. A packet is therefore analogous to a time-slot in TDM, although packets are typically much larger (hundreds or thousands of bits as opposed to one or eight). Packets can contain a fixed number of information bits, the same for each packet, or more often the number can be variable from one packet to another (usually with a maximum). The basic idea is then to interleave the packets from different users on the communications link, not unlike in TDM except that by not pre-determining the order of packets from different users or the size of packets we can readily vary dynamically the bit rate assigned to each user.

**Example 18-15.**

End users sitting at terminals and communicating to central computers typically type a line of characters, followed by a "carriage return", and then expect some response from the computer. Thus, it would be natural to form a packet of user information bits corresponding to this line of characters plus carriage return. □

**Example 18-16.**

Conversational speech consists of active speech intervals interspersed with silence intervals. It would be natural to associate a packet with each active interval. In practice, since active intervals can get quite long, we can associate two or more packets with a single interval of active speech. □

In TDM we identified at the output of the link the individual time-slots corresponding to different users by adding framing bits and doing framing recovery at the receiver. We must realize the analogous function in packet switching, although by a somewhat more flexible mechanism. First recognize that since the bit rates provided to each users is varying dynamically, we have a problem if the sum of the incoming bit-rates is instantaneously higher than the total bit-rate of the communications link. We will defer this problem until the next subsection, and for the moment assume that the incoming bit-rates sum to less than the link bandwidth. This implies that there must be *idle-time* on the link. During this idle-time, by mutual convention between

transmitter and receiver, we transmit a deterministic sequence, say all-zeros. We must then have some way of identifying at the receiver the beginning and end of a transmitted packet. Since the beginning and end of packets do not occur at predetermined points in time, we say that the packets are *asynchronous*. The process of determining the beginning and end of packets at the receiver is called *synchronization* to the packets, and is analogous to framing recovery in TDM.

For purpose of synchronization, we append to the beginning and end of a packet additional bits, called *synchronization fields*. The combination of the packet, together with the synchronization field and other fields yet to be described is actually the unit of bits transmitted on the link, called a *link frame*. Thus, this frame is analogous to the frame defined in TDM, except that in TDM the frame corresponds to bits from *all users*, whereas in packet switching it corresponds to the bits from a *single user*. The synchronization fields are entirely analogous to added framing bits in TDM, since they provide a deterministic fixed reference in the bit stream for the start and end of the bits corresponding to one user. The term *field* applies in general to any fixed-length collection of bits added to the information packet to form the link frame. We will see examples of other fields shortly.

#### Example 18-17.

An internationally standardized packet switching format is *high-level data link control* (HDLC) [5,6]. Many specific formats are subsets of HDLC, such as the common international packet switching interface standard X.25. HDLC uses one synchronization field at the beginning and one at the end of a link frame. Each field is called a *flag*, and consists of the octet "01111110". Since by convention an idle link contains all-zeros, it is easy to recognize the beginning of a link frame by observing this particular octet. However, we still have a problem with reliably detecting the end of a link frame, since the octet "01111110" may occur in the information packet, thus prematurely ending the link frame. To bypass this problem, we use a form of *variable-rate* coding similar to that discussed in Section 12.1.5. We simply insert within the packet any time that five consecutive one-bits appear, an additional zero-bit. Thus, within the packet, before appending the flags to form a link frame, we use the encoding rule

$$11111 \rightarrow 111110$$

and at the demultiplexer we apply the decoding rule

$$111110 \rightarrow 11111$$

to recover the original packet. Note that this *bit-stuffing encoding* increases the number of bits in the packet by the maximum ratio 6/5 or 20%. This is the price we pay for avoiding the replication of the end flag in the information packet. This scheme allows a variable number of bits per packet, since there is no presumption of the length of a packet. In fact, the variable-rate encoding *ensures* that the link frames, if not the packets, will be variable length, since the number of added coding bits will depend on the information bits.  $\square$

Of course in the presence of bit-errors on the link, we can lose synchronization of the beginning and end of a link frame (Problem 18-3). In this eventuality we must have a *recovery procedure* analogous to framing recovery in TDM.

Once we have synchronized to the beginning and end of a link frame at the receiver, there is still need to identify the user corresponding to each packet. For this

purpose, we include the concept of a *connection*, where one user (a human being or computer) may have multiple connections. In TDM each connection corresponded to a time-slot, and the identification of connections was implicit in the location of a time-slot within the frame. In packetizing we usually also identify the concept of a connection, but the flexibility of the packetizing approach results in no fixed correspondence between connections and the sequence of packets. Therefore we must add an *address field* to the link frame. Each connection through a link is assigned a unique address, and the size of the address field places a restriction on the number of simultaneous connections (in contrast to TDM where the number of time-slots in a frame limits the number of connections).

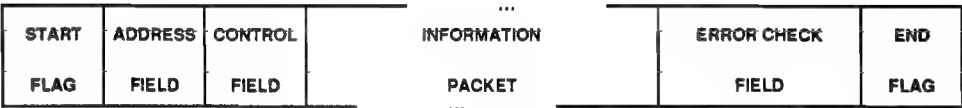
**Example 18-18.**

For HDLC in Example 18-17, an address field of one octet immediately follows the start flag. Thus, there are 256 possible simultaneous connections. The full HDLC link frame is shown in Figure 18-7. There are two additional fields, the control and error check fields. Each packet has appended six octets or 48 bits to form the link frame. Obviously, if only very short packets are transmitted, the overhead is substantial. On the other hand, if the packets average thousands of bits, the overhead as a fraction of the entire link bandwidth is insignificant. □

**Statistical Multiplexing**

We have seen a packetizing technique for sharing a link among a set of connections in a much more flexible manner than in TDM. However, we have ignored some important problems that are brought out when we consider the design of a multiplexer that takes a number of lower-speed links, each packetized, and multiplexes them together in a higher-speed link. This device is called a *statistical multiplexer*, and is the packetizing equivalent of a TDM multiplex.

The two problems that must be addressed are the probability of two or more packets arriving simultaneously at the statistical multiplexer, or more generally the possibility that since each connection has a variable bit rate, the total incoming bit rate exceeds the bandwidth of the link. If the latter condition persists indefinitely, since the multiplexer has a finite internal memory it is inevitable that some packets must be *lost*. The probability of this occurring can be minimized by using some sort of *flow control*, which is a mechanism for telling the originators of the packets that they must reduce their bit rate due to over-utilization of some link.



**Figure 18-7.** The data link frame defined for HDLC.

Even in the absence of over-utilization in the long-term sense, it is inevitable that the instantaneous bit rate into the multiplex will sometimes exceed the total link bit rate out of the multiplex. If this were not true, the output bit rate would exceed the sum of the maximum bit rates into the multiplex, in which case we would consider using the simpler TDM. This leads to the conclusion that the multiplex must include internal buffer memory for storage of the excess bits. Statistical multiplexing makes more efficient use of the link bandwidth by reducing the output bit rate below the maximum instantaneous sum of incoming bit rates. When the input bit rate exceeds the output rate, it stores the excess bits until the input bit rate is subsequently lower than the output, at which time it transmits the bits in memory. It therefore takes advantage of the *statistics* of the variable-rate inputs to make more efficient use of the output link. The price we pay for this efficiency is the *queueing delay* that is an inevitable consequence of temporarily storing packets in internal buffers before transmission. This delay is statistical in nature, so that we must speak in terms of the average delay, the distribution of the delay, and so forth. The impact of this delay depends on the service that is being offered by packetizing.

**Example 18-19.**

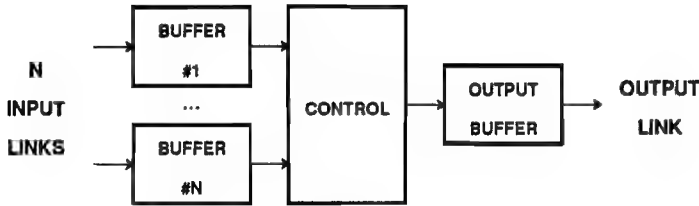
In interactive data transmission, the queueing delay results in a slower response time from a remote computer. This delay can be disturbing if it gets to be large. □

**Example 18-20.**

Using packetizing for speech transmission results in a random change in the temporal characteristics of the speech unless we take additional measures. One approach is to add to each packet a *time-stamp* which indicates roughly the time elapsed since the end of the last packet (i.e. duration of the silence interval). At the receiver we can restore the rough temporal relationship by adding another buffer that forces a constant rather than variable delay. Based on knowledge of the currently experienced delay variation, this buffer adds a delay to each packet so that the total delay (queueing delay plus buffer delay) adds to a constant. This constant delay will be subjectively better than a variable delay, although it introduces its own conversational and echo problems. □

We always have to cope with lost packets in statistical multiplexing. This is because with some non-zero probability, the instantaneous input bit rate will exceed the capacity of the output link for a sufficiently long period of time that the finite buffer capacity of the multiplex will be exceeded, and bits will be lost.

With these concepts in mind, the statistical multiplex is easy to understand, and is illustrated in Figure 18-8 [7]. Associated with each input link to the multiplex is an input buffer which stores packets as they arrive. The remainder of the link frame, with the exception of the address, can be discarded. The address must be retained and attached to the information packet on the output link. Separate buffers are required because packets may be arriving simultaneously on the input links. A control unit examines the input buffers for packets, and when it finds them places them in the output buffer in the order in which they are to be transmitted. Strictly speaking, the output buffer would not be required, since packets could be removed from the input buffers and transmitted directly. However, the use of an output buffer allows the



**Figure 18-8.** Block diagram of a statistical multiplexer.

control unit more freedom in its operation, and also conveniently keeps the packet in memory so that it can be retransmitted if it is lost (should the protocol in use dictate that). Each of the buffers is a *first-in first-out (FIFO) buffer*, meaning that the packets are read out in the order they arrived at the buffer input.

The designer of the multiplex must decide on some strategy for moving packets from input buffers to output. This is called the *queue management discipline*, and affects the distribution of queueing delays experienced by packets passing through the multiplex. The most obvious discipline would be to take the packets out of input buffers in order of arrival, called a *first-in first-out discipline*. The FIFO discipline would make the whole multiplex equivalent to a single FIFO queue, which is simple to analyze. However, this discipline is difficult to implement, since each packet stored in a buffer would have to have an associated time-of-arrival stamp, and the control unit would have to examine this stamp for the oldest packet in every buffer prior to choosing one for transmission. A much more practical discipline is *roll-call polling*, in which the control unit systematically goes through the buffers in order, removing all waiting packets at each buffer. Other disciplines are possible (Problem 18-4). In the evaluation of different strategies, the primary considerations would be implementation and the delay characteristics. Most disciplines are very difficult to analyze, and occasionally we must resort to simulation.

In order to illustrate the essential delay characteristics of a statistical multiplex, we will analyze the discipline that is the most analytically tractable. This is the FIFO discipline, because in this case the multiplex is equivalent to a single FIFO queue, for which we can directly use the analytical results in Chapter 3. For purposes of calculating the delay, two characteristics are important other than the queue management discipline. One is the process modeling the arrival of packets at the multiplex, and the other is the distribution of service times. For a statistical multiplex, the service time is the time required for transmission of the packet, which for a fixed bit rate output link is proportional to the length of the packet. The analytically simplest case is the M/M/1 queue analyzed in Section 3.4, where the packet arrivals are Poisson, the queue has an unlimited number of waiting positions, and the service time is exponentially distributed. Let the *total* arrivals on all  $N$  input links be a Poisson process with arrival rate  $\lambda$ , and let the service time have mean-value  $1/\mu$  (the service rate is  $\mu$ ). The quantity



$$\rho = \lambda/\mu \quad (18.1)$$

is the average server utilization, or in this case *link utilization*; that is, the average fraction of the time that the output link is transmitting packets. Obviously we would be happiest if this quantity were as close to unity as possible, but as we will see this leads to large queueing delays. The average queueing delay, or the average time a packet sits in the buffer before beginning transmission, is given by (3.131),

$$D = \frac{1}{\mu} \cdot \frac{\rho}{1 - \rho} \quad (18.2)$$

The term  $1/\mu$  is the average transmission time, and the queueing delay can be smaller (when  $\rho < 1/2$ ) or larger (when  $\rho > 1/2$ ). The important point to realize is that the queueing delay gets very large as the output link utilization approaches unity. This places an upper bound on the link utilization that can be achieved. Intuitively, by keeping the utilization low, we enhance the probability that the link will be free when a packet arrives.

## Packet Switching

The statistical multiplex can be modified to provide a switching function as well. A *packet switch* has an equal number of input and output links. The function of the switch is to route packets on each incoming link to the appropriate output link. Thus, the switch maintains an internal table which translates from an address and input link number to a corresponding output link and associated address. The internal configuration of the packet switch is similar to Figure 18-8, except that there are  $N$  output buffers and  $N$  output links.

In practical implementations packet switching involves many complexities that we have barely touched on here. There are many complex procedures for establishing and taking down connections and for recovery from every conceivable kind of error. These procedures plus the details of the link frame constitute the set of *protocols* used to form a complete communications network based on packet switching. Packet switching has many advantages over circuit switching to exchange for its greater complexity. In addition to the ones mentioned earlier, a particularly important distinction is that packet switching encourages the user to establish many simultaneous virtual connections through a communications network. As such, it is an extremely flexible approach for situations in which users wish to access many computational or peripheral resources simultaneously or in quick succession.

### Example 18-21.

In an office environment, many users may wish to access the same printer or storage device. Packet switching encourages this, because it allows that device to establish simultaneous connections to many users. □

## 18.2.4. Packet Switching Multiple Access

Thus far we have discussed packet switching in the context of an important special case of multiple access, the link topology. It is applicable more generally, and in fact most of the practically useful multiple access techniques are based on packet

transmission. The techniques are distinguished mainly by the degree of *centralized* vs. *distributed control* and by their *efficiency* and *stability*. Efficiency refers to the degree to which they can successfully use the available bandwidth on the multiple access medium, since they all require some overhead and idle time, and stability refers to the possibility that during high utilization the frequency of lost packets can increase uncontrollably.

### Collision Avoidance by Polling

In the typical multiple access situation, as opposed to the statistical multiplex, the nodes transmitting on the medium are distributed, and have significant propagation delays between them. Each node can be considered to contain a buffer, in which packets are queued awaiting transmission. The goal is to allow the nodes to transmit their packets, but to coordinate the transmissions so that they do not collide. In TDMA we accomplished this through the relatively inflexible time-slot assignment. By using packetization approaches, we can allocate the medium bandwidth much more dynamically among the nodes.

We already mentioned in connection with the statistical multiplex a valuable approach directly applicable to this problem — *polling*. Polling comes in two forms, *roll-call polling* which is managed by a central controller, and *hub polling* which is more distributed. In roll-call polling, a central controller sends a message to each node in turn, letting it know that it is allowed to transmit. That node then either transmits the packets waiting in its buffer (or perhaps only one packet maximum, depending on the discipline), or sends back a message indicating that its buffer is empty. Every node is aware of the status of the medium at any given time, and in particular which node (including the central controller) is currently authorized to transmit. There is therefore no possibility of a collision on the medium.

#### Example 18-22.

Roll-call polling is widely used in dispersed networks of voiceband data modems accessing a centralized computer facility, as for example an airline reservations system. The wide use of polling techniques with voiceband data modems motivates the desire for these modems to acquire timing and carrier quickly, and has led to a lot of research in fast acquisition. As will be shown in a moment, reducing the acquisition time, and therefore the overhead in polling, enhances the performance of the polling technique. □

Hub polling, called *token-passing* in the context of local-area networks, eliminates the central controller except for initialization. In this case, a token is possessed by precisely one node in the network at any given time. This token is not a physical object, but rather an authorization to transmit on the medium. The node possessing the token can transmit, and at the end of that transmission must pass the token on to a predetermined next node (the token is passed through a message transmitted on the medium). In this fashion, as the token is passed among all the nodes each has an opportunity to transmit its packets.

**Example 18-23.**

A popular LAN architecture is the *token-passing ring*. The topology is a ring, and the token is passed around the ring from one station to another. The technique is relatively simple because each node passes the token to its nearest neighbor on the ring by transmitting a generic message that is intercepted and removed by the next node. There is no need to know the details of who the nearest neighbor is, its address, etc.  $\square$

We can understand polling techniques better by performing a simple calculation of the *average polling cycle time*, or the time required for each node to be given the opportunity to transmit its packets. Assume that the total overhead time for one polling cycle is  $W$  — this includes the time for the messages to pass from the controller to nodes and back, or the time to transmit the tokens. Then we get that the polling cycle time is

$$T_c = W + \sum_{i=1}^N T_i \quad (18.3)$$

where  $T_i$  is the time it takes node  $i$  to transmit all its packets when polled. Taking the expected value of this,

$$E[T_c] = W + N \cdot E[T_i] \quad (18.4)$$

assuming that each node has the same utilization. If each node has utilization  $\rho$ , that is it transmits a fraction  $\rho$  of the time, then considering that each node transmits the packets accumulated during one polling cycle,

$$E[T_i] = \rho \cdot E[T_c] \quad (18.5)$$

and combining these two equations we get

$$E[T_c] = \frac{W}{1 - N \cdot \rho} \quad (18.6)$$

As expected, as the total utilization  $N \cdot \rho$  approaches unity, the polling cycle lengthens, and as it approaches zero the polling cycle time approaches the overhead  $W$ . The fraction of the polling cycle devoted to overhead is

$$\frac{W}{E[T_c]} = 1 - N \cdot \rho \quad (18.7)$$

This equation illustrates a very important advantage of polling; namely, as the total utilization approaches unity, the fraction of the time devoted to overhead decreases to zero. Thus, the overhead is only appreciable when the utilization is low, a situation in which we don't care, and is insignificant when the utilization is high, which is precisely what we would hope. Intuitively this is because during high utilization the polling cycle is very long, and the fixed overhead becomes insignificant as a fraction of this cycle.

An undesirable feature of polling is that the overhead time  $W$  increases with the number of nodes in the network. Thus, it becomes inefficient for networks with a very large number of nodes, each with low utilization of the medium. In this situation the polling cycle becomes dominated by the overhead. For these types of networks, random access techniques as described in the following subsection are very desirable

because they can obtain comparable performance but without the complexity of the centralized control.

### Random Access

Thus far in this chapter the focus has been on *avoiding* collisions in multiple access to a common medium. On media that are used with a low utilization, the complexities associated with avoiding collisions can be circumvented by a strategy of allowing collisions to occur, detecting those collisions, and retransmission with a random delay. This enables each node of the network to operate autonomously, with no central control required.

The first and simplest such strategy was invented by N. Abramson of the University of Hawaii in 1970 [8] and is known as *pure ALOHA*. ALOHA is appropriate for broadcast topologies, such as the bus or satellite, where collisions are easy to detect because each node listens to all transmissions including its own. If a node cannot successfully monitor its own transmission packet, it can assume that a collision has occurred. The technique is then very simple: each node simply transmits a packet as desired, regardless of conditions on the medium, and monitors the medium for a collision. When a collision occurs, the node waits for a random delay time, and retransmits. The random delay time makes it less probable that the retransmissions of the colliding nodes will again collide.

We can analyze this system easily if we make a simplifying assumption. Even if the incoming packets to the system have Poisson arrivals, we would not expect that the aggregate of packets on the medium, including retransmitted packets, would be Poisson. However, we assume this to be the case, yielding an approximate analysis that has proven on further examination to be accurate as long as the random retransmission times are long relative to a packet length. Further assume that packets are a *fixed length*  $1/\mu$ , for simplicity of calculating the probability of a collision. Let the total rate of arrivals of packets to the system be  $\lambda_{in}$  and let the rate of arrivals of packets on the medium including retransmissions be  $\lambda_{out} > \lambda_{in}$ . Due to the fixed-length packet assumption, if we transmit a packet at time  $t_0$ , then a collision occurs if someone else transmits a packet in the interval  $[t_0 - 1/\mu, t_0 + 1/\mu]$ . Due to the Poisson assumption, the probability of *no* collision is the probability of zero Poisson arrivals for a Poisson process with rate  $\lambda_{out}$  over an interval of time  $2/\mu$ , or  $\exp\{-\lambda_{out} \cdot 2/\mu\}$ . But the probability of no collision is also the ratio of the rate of incoming packets to packets on the medium,  $\lambda_{in}/\lambda_{out}$ , since the excess are retransmissions due to collisions. Setting these two expressions equal, we get

$$\frac{\lambda_{in}}{\lambda_{out}} = \exp\{-\lambda_{out} \cdot 2/\mu\} \quad (18.8)$$

It is convenient to express this in terms of the utilization due to incoming packets and the total utilization of the medium,

$$\rho_{in} = \frac{\lambda_{in}}{\mu} \quad \rho_{out} = \frac{\lambda_{out}}{\mu} \quad (18.9)$$

in which case we get the interesting relation

$$\rho_{in} = \rho_{out} e^{-2\rho_{out}} . \quad (18.10)$$

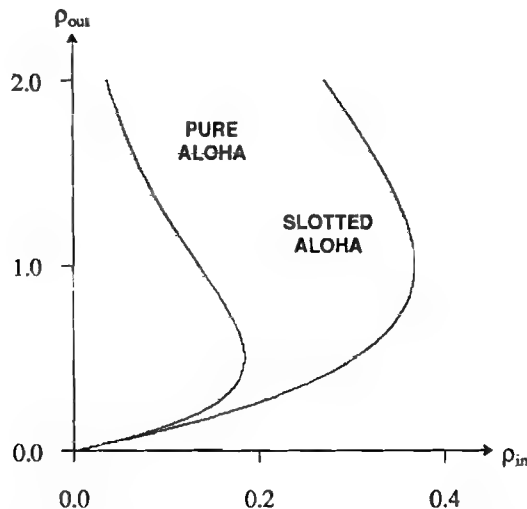
In this equation,  $\rho_{in}$  is the independent variable, the incoming traffic, and  $\rho_{out}$  is the dependent variable, the traffic on the medium. This implicit relation is plotted in Figure 18-9 with the independent variable on the abscissa. Note that the utilization of the medium can exceed unity, as we would expect in the case of a large number of collisions and retransmissions. What is most interesting about the curve is that the incoming traffic can never exceed 0.18, or in other words the random access technique can only work for very low utilizations.

#### Exercise 18-2.

Show that the maximum input traffic in Figure 18-9 corresponds to  $\rho_{in} = 1/2e = 0.18$  and  $\rho_{out} = 1/2$ .  $\square$

Another interesting feature is the double-valued nature of the curve — for each  $\rho_{in}$  in the allowed region, there are *two* operating points possible, one with a small number of collisions and the other with a large number of collisions. This implies a form of *instability*, since the system performance cannot be predicted unambiguously.

Since the original concept of pure ALOHA, a great deal of effort has been expended to increase the throughput of random access techniques and to ensure stability. A simple refinement, known as *slotted ALOHA* (Problem 18-7) results in a doubling of throughput to 0.37. A class of control algorithms known as *collision-resolution algorithms*, first conceived by J. Capetanakis at M.I.T. [9] can ensure



**Figure 18-9.** The throughput of the pure ALOHA discussed in the text and the slotted ALOHA derived in Problem 18-7.

stability on a random access channel. The maximum throughput that can be achieved on such a channel is known only to fall in the range of 0.45 to 0.59. Many have speculated that it must be 0.5. See [10] for an excellent review of results on this topic.

## CSMA/CD

Random access protocols are widely used in local-area networks, although using a modification of the ALOHA system in which collisions are largely (but not entirely) avoided. The most common approach is for each node to first listen to the medium to determine if a transmission is in progress before proceeding to transmit. This is known as *carrier-sense multiple access (CSMA)*. This greatly reduces the probability of a collision, but does not rule it out because due to propagation delays it is not always possible to detect that another node has just started a transmission. It is therefore common to use *collision-detection (CD)* by listening to one's own transmission, and if a collision is detected to transmit a special *jam signal* that serves to notify other nodes to that effect and then suspend transmission. A substantial improvement in throughput is obtained by reducing the number of collisions and by aborting transmissions in progress when a collision occurs.

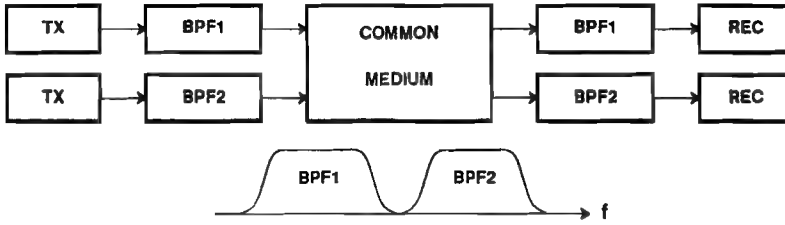
### Example 18-24.

CSMA/CD schemes differ in the retransmission strategy once a collision occurs [6]. The best-known CSMA/CD system is the popular *Ethernet* local-area network, usually implemented on a coaxial cable at a bit rate of 10 Mb/s. In Ethernet [11] after a collision is detected and transmission is aborted, a node retransmits after a random delay. This retransmission interval is doubled upon each successive collision, up to some maximum. □

## 18.3. MULTIPLE ACCESS BY FREQUENCY DIVISION

The separation of many users on a common medium in analog transmission invariably uses frequency division. For example, on the familiar AM radio dial, the stations are selected by tuning a variable frequency filter. Frequency division can also be used for digital transmission, where independent data streams are transmitted in non-overlapping frequency bands. This occurs most commonly when data streams are transmitted over existing analog carrier systems. But it is also used when data streams are transmitted by radio or satellite, and for full duplex data transmission.

Frequency-division multiplexing is very simple, as illustrated in Figure 18-10, in this case for just two users. The two transmitters sharing the medium have output power spectra in two non-overlapping bands, where they usually use passband PAM modulation to achieve this. To ensure that this is the case, it is common to put bandpass filters at the output of the transmitters, particularly where strict requirements on spectral utilization are in force (as in the case of the FCC mask for terrestrial microwave radio in Figure 5-21). At the two receivers, similar bandpass filters eliminate all but the desired data signals. The path to a receiver from the undesired transmitter contains two bandpass filters with non-overlapping passbands, and therefore we can make the loss of the crosstalk path as large as we like through the filter



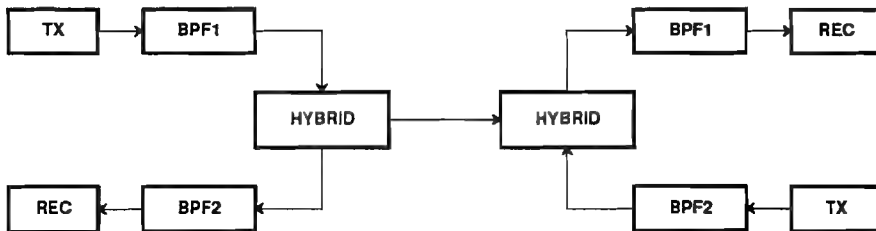
**Figure 18-10.** Sharing a medium between two users using frequency-division multiplexing.

design. A consideration in many microwave media (terrestrial radio and satellite) is nonlinearity of the amplifiers in the transmitter or the satellite transponder. These nonlinearities will create out-of-band energy that can interfere with other FDM channels, and it is common to use bandpass filtering *after* the amplifier to reduce these spectral components.

FDM has both advantages and disadvantages over TDM. A major disadvantage on all media is the relatively expensive and complicated bandpass filters required, whereas TDM is realized primarily with much cheaper logic functions. Another disadvantage of FDM is the rather strict linearity requirement of the medium. However, some of the propagation-delay and crosstalk problems of TDM are eliminated by FDM. The technique of choice then depends on the situation.

#### Example 18-25.

On microwave radio channels, FDM techniques have been used primarily, but TDM is in the process of taking over. The primary advantage of TDM, and particularly TDMA in the case of the satellite channel, is the lowered susceptibility to nonlinearity of amplifiers. Using TDM, the amplifiers can be used nearer saturation, giving a greater available power. The reduction of filtering requirements is an added benefit. □



**Figure 18-11.** Frequency-division multiplexing of two directions in full-duplex transmission.

**Example 18-26.**

In full-duplex data transmission, TDM in the form of TCM is impractical on media with long propagation delays, such as for voiceband channels. Lower-speed voiceband data modems therefore use FDM for full-duplex transmission, as shown in Example 18-26. The hybrids turn the two-wire medium into an effective four-wire medium, and the stopband loss of the bandpass filters is added to the hybrid loss resulting in good isolation of the two directions. FSK modulation is used to generate a passband signal, with two different carrier frequencies used for the two directions, at 300 b/s, and passband PAM (PSK and QAM) is used at higher frequencies. □

**Example 18-27.**

For the digital subscriber loop, FDM has the major advantage of eliminating near-end crosstalk, and is not susceptible to propagation delay like TCM. However, it is not used because of the complicated filtering and the larger bandwidth (as compared to echo cancellation, Chapter 19) on a medium with an attenuation increasing rapidly with frequency. □

**Example 18-28.**

In direct detection optical fiber systems, FDM in the form of *wavelength-division multiplexing* is sometimes used. A few channels are separated by transmitting them at different wavelengths. In the future, coherent optical fiber will encourage the use of FDM multiplexing of large numbers (hundreds or thousands) of digital streams. This looks like a very promising use of FDM. □

## 18.4. MULTIPLE ACCESS BY CODE DIVISION

Separating signals in time or frequency is a relatively simple way to ensure that the signals will be orthogonal. However, it is by no means the only way. In this section we briefly describe the design of orthogonal signals by *code division*, which is closely related to spread-spectrum (Sections 6.7 and 8.6), and was previously described in Section 6.9.2.

With *code-division multiple-access (CDMA)*, the objective is to transmit signals from multiple users in the same frequency band at the same time. We can place this alternative in perspective by comparing it to TDMA and FDMA. Suppose we have  $B$  Hz of available bandwidth, and want to share this bandwidth over  $N$  users. Several options are available:

- **TDMA.** A single stream of PAM pulses with symbol rate  $1/T = B$  can be transmitted. (We assume throughout passband PAM with maximum symbol rate consistent with the Nyquist criterion.) We can divide the stream of pulses into  $N$  time slots, assigning each timeslot to one of the  $N$  users, so that each user has available a net symbol rate of  $1/NT = B/N$ .
- **FDM or FDMA.** Divide the frequency band  $B$  into  $N$  disjoint frequency bands, and assign one to each user. Each user then gets a frequency band  $B/N$  Hz wide, and the highest symbol rate that each user can achieve is  $1/T = B/N$ , the same as



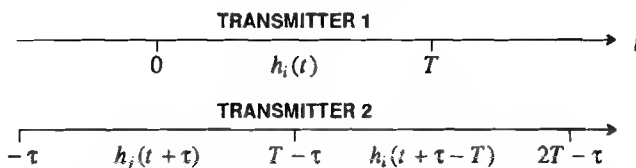
with TDMA.

- **CDMA.** In accordance with the generalized Nyquist criterion, design  $N$  orthogonal pulses. These pulses each satisfy the Nyquist criterion with respect to symbol interval  $T$ , and are mutually orthogonal for all translates by multiples of  $T$  sec. A requirement of the generalized Nyquist criterion is that  $B \geq N/T$ . Each user is assigned one of these orthogonal pulse shapes to use within bandwidth  $B$  Hz, and transmits with symbol rate  $1/T$  Hz. The receiver for each user consists of a sampled matched filter, which has no ISI at the output and does not respond to the orthogonal pulses. The maximum symbol rate that each user can achieve is  $1/T = B/N$ , the same as TDMA and FDMA.

Each of these approaches can achieve, in principle, the same aggregate spectral efficiency, since each one achieves the same symbol rate per user, the same number of users, and the same total bandwidth. Each approach has its advantages and disadvantages. In this section, we concentrate on CDMA.

As shown in Section 8.6, pulses with large bandwidth relative to the symbol rate can be generated using a combination of a chip waveform and a spreading sequence. This is known as *direct-sequence spread spectrum*. Further, it was shown in Chapter 12 that maximal-length shift register sequences have the appropriate properties to be used as spreading sequences, and are very easy to generate in the transmitter and receiver. For CDMA, we can assign one period of a maximal-length shift register sequence to each user to use as their spreading sequence. All users use the same chip waveform. (This constrains the spreading factor  $N$  to be of the form  $2^n - 1$  for some integer  $n$ . By assigning different generator polynomials to each user, orthogonality can be achieved. An example of a set of generator polynomials that achieve orthogonality is the *Gold code* [12].

In principle CDMA achieves the same spectral efficiency as TDMA or FDMA, but in practice there are factors that give it much different characteristics. Two considerations are the *near-far* problem, and the *partial correlation* problem. Partial correlation arises where no attempt is made to synchronize the transmitters sharing the channel, or when propagation delays cause misalignment even when transmitters are synchronized. This is shown in Figure 18-12, where the symbol interval for one transmitter is delayed by  $\tau$  seconds relative to the other. In order to avoid crosstalk between the two users  $i$  and  $j$ , two *partial correlations* must be zero,



**Figure 18-12.** Illustration of the phase relationship between symbol intervals of two users of a CDMA system.

$$\int_0^{T-\tau} h_i(t) h_j(t + \tau) dt = 0 \quad (18.11)$$

$$\int_{T-\tau}^T h_i(t) h_j(t + \tau - T) dt = 0 \quad (18.12)$$

Thus, simple orthogonality of the isolated pulses is not sufficient — these two partial correlations must also be zero, or at least small, for any value of  $\tau$ . This property is not guaranteed by the generalized Nyquist criterion of Chapter 6, which assumes a known relationship between the different symbol intervals. The partial correlations can be reduced by proper choice of the spreading sequences, but cannot be totally eliminated. This implies that in practice, complete orthogonality of the different pulses assigned to different users cannot be maintained. CDMA system capacity is thus typically limited by the interference from other users, rather than by thermal noise.

The near-far problem is analogous to near-end crosstalk on wire-pair media, and results when geographically dispersed users are sharing a common medium such as a radio channel. If all the users transmit at the same power level, then the received power is higher for transmitters closer to the receiving antenna. Thus, transmitters that are far from the receiving antenna are at a disadvantage with respect to interference from other users. This inequity can be redressed by using *power control*. Each transmitter can accept central control of its transmitted power, such that the power arriving at the common receiving antenna is the same for all transmitters. In other words, the nearby transmitters are assigned a lower transmit power level than the transmitters far away.

When there are  $N$  total users in a CDMA system, then from the perspective of one particular transmitter there are  $N - 1$  interferers. If power control is used, then each transmission arrives at the receiver with the same power  $S$ . The SNR is defined as the ratio of signal power to total interference power, and is

$$SNR = \frac{S}{(N - 1)S} = \frac{1}{N - 1} \quad (18.13)$$

By the central limit theorem, the interference, consisting of a superposition of independent transmissions, will be approximately Gaussian. For large  $N$  the SNR is very poor, and one might expect unreliable system operation. However, as we saw in Section 8.6, it is not the SNR that matters when a matched filter receiver is used, but rather the ratio of the signal energy per symbol to the interference spectral density. The signal energy per symbol interval is  $T \cdot S$  for symbol interval  $T$ . If we model the interference signal as white Gaussian noise, then it has total power  $(N - 1)S$  and bandwidth  $2B$  (for positive and negative frequencies), and hence has power spectrum  $N_0 = (N - 1)S/2B$ . Thus, the ratio of energy per symbol to noise density is

$$\frac{\sigma_h^2}{N_0} = \frac{2BT}{N - 1}, \quad (18.14)$$

which reflects the spread-spectrum processing gain  $2BT$  (Section 8.6). Thus, for a given number of users  $N$ , by making  $2BT$  large enough we can always make  $P_e$

sufficiently small. The minimum value,  $2BT = N$  (in accordance with the generalized Nyquist criterion), results in a poor  $P_e$ , but we can always make  $2BT$  larger than this minimum.

Processing gain is important for CDMA as well as spread spectrum. For  $N$  users, where the received power of each user is fixed at  $S$  through power control, the  $N - 1$  interferers represent a signal analogous to the jamming signal discussed in Section 8.6. The total power of the interference stays constant as we increase the bandwidth (processing gain), and hence the spectral density of the interference decreases. This decrease in spectral density results in a  $P_e$  that decreases as the bandwidth (and processing gain) increases.

Since  $2BT \gg N$  in a CDMA system with a small  $P_e$ , it appears that CDMA systems have a lower spectral efficiency than TDMA or FDMA systems, which achieve close to  $2BT = N$ . However, this perspective is oversimplified for many multiple access applications, as illustrated by some examples.

#### Example 18-29.

CDMA has been proposed for the North American digital cellular telephone network, and a *higher* capacity has been estimated for CDMA than TDMA or FDMA [13,14]. In part, these estimates are based on another factor, voice activity. The transmitter is activated only during periods when the user is talking, and it can be assumed that some fraction of the users are talking at any given time. For a given acceptable interference power (based on the maximum  $P_e$ ), the number of telephone users can be increased if each user is transmitting only a portion of the time. TDMA and FDMA systems can also take advantage of voice activity, but only by much more complicated mechanisms. Another factor in favor of CDMA is cellular frequency assignment, as discussed in Section 18.5.  $\square$

#### Example 18-30.

On a local area network (LAN), each station is typically only transmitting a portion of the time. The CDMA capacity is expressed in terms of the number of active stations at any given time, not the number of total stations. CDMA is advantageous for this type of network because it requires no synchronization of the multiple communications sessions occurring at any given time. It also naturally takes advantage of the low duty cycle of transmission of the stations.  $\square$

## 18.5. THE CELLULAR CONCEPT

Thus far in this chapter, we have discussed multiple access by frequency, time, and code division. There is a fourth alternative, which is *space division*. On cables and fibers, this is manifested by parallel communications on physically separate cables or fibers. A more interesting manifestation of space-division multiple access is the *cellular concept* used in mobile radio systems.

In radio systems where a large geographic coverage is desired and large numbers of mobile transceivers must be supported, it is common to divide the region into cells. This is illustrated in Figure 18-13, where a regular array of base stations (including

transmitter and receiver) is deployed, each one dedicated to mobile users in its immediate area. Each mobile transceiver communicates with the nearest base station, resulting in hexagonal regions associated with each base station as shown in Figure 18-13a. The motivation is to allow the same carrier frequency to be re-used in different cells, increasing the overall system capacity. Transceivers in two different cells can be assigned the same carrier frequency and time slot, providing that the two cells are far enough apart, since the remote transceiver will suffer a much larger propagation loss. A given mobile transceiver will pass through a succession of cells, as shown in Figure 18-13b. As it passes from one cell to the next, it must establish communication with the base station associated with the new cell, possibly requiring a reassignment of carrier frequency and timeslot.

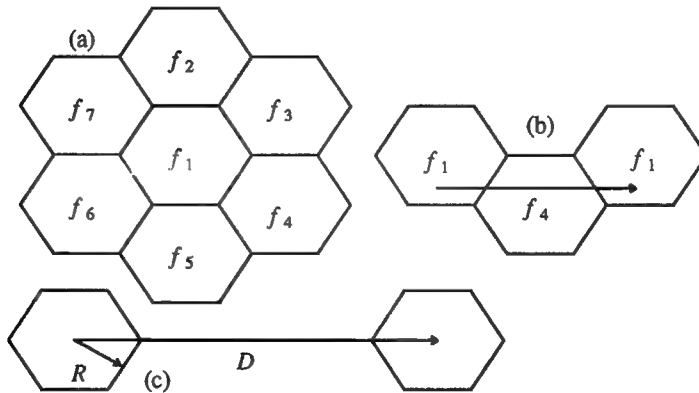
The key parameter of a cellular configuration and frequency assignment is the ratio of  $D$  to  $R$ , where  $D$  is the distance to the nearest cell that uses the same carrier frequency and  $R$  is the diameter of one cell (see Figure 18-13c). The larger the ratio  $D/R$ , the lower the interference from nearby cells assigned the same carrier frequency.

#### Example 18-31.

If in a radio system, the propagation obeys the fourth-power law, then given two mobile transmitters with the same transmit power, the desired transmitter at maximum distance  $R$  and the interfering transmitter at distance  $D$ , then the signal-to-interference ratio is

$$\frac{R^{-4}}{D^{-4}} = (D/R)^4, \quad (18.15)$$

or  $40 \log_{10} D/R$  dB. Thus, the larger  $D/R$ , the smaller the interference in relation to the



**Figure 18-13.** Cellular mobile radio uses a regular hexagonal array of transmitters. (a) The total coverage area is divided into hexagonal cells. (b) A given mobile user will pass through a succession of cells. (c) Each cell has radius  $R$ , and is located at a distance  $D$  from the nearest cell assigned the same carrier frequency.

signal. The tradeoff is 12 dB increase in signal-to-interference ratio for each doubling of the  $D/R$  ratio.  $\square$

One goal in designing a modulation scheme for a cellular radio system is to decrease the allowable  $D/R$ . All else being equal, this will increase the total system capacity by allowing the same frequency to be assigned to cells closer to one another. Another goal is to design a modulation and multiple access scheme within each cell that is least susceptible to interference from other cells.

**Example 18-32.**

If FDMA is used, generally the same carrier frequency cannot be assigned to users in adjacent cells. This is because two transmitters using the same carrier frequency can be very close to one another at the boundary between the cells, and cannot be separated at their respective base station sites. However, it is generally permissible to assign the same carrier frequency to cells that are not adjacent, presuming that a  $D/R$  ratio on the order of three is permissible. Thus, there is typically a seven-cell frequency reuse pattern, where each carrier frequency is assigned to only one out of each cluster of seven adjacent cells.  $\square$

**Example 18-33.**

In the North American IS-54 digital cellular standard for mobile telephony, a combination of FDMA and CMDA is used. Each carrier frequency is modulated with a bit stream that is in turn divided into three (or ultimately six) time slots. Three different users are assigned distinct time slots on that single carrier. The same carrier frequency can be assigned to transceivers in adjacent cells, providing they are assigned to non-overlapping time slots. In this fashion, there will be no interference from adjacent cells.  $\square$

**Example 18-34.**

If CDMA multiple access is used, then transceivers *within* cells can be assigned the same carrier frequency and time slot, but are assigned distinct spreading codes. It follows that each carrier frequency can be reused in adjacent cells. The transceivers in adjacent cells manifest themselves as an increase in interference power that can be compensated by increasing the processing gain. That increase in processing gain is reduced by the greater distance to transmitters in adjacent cells. Like the voice activity factor (Section 18.4), this complete frequency reuse in adjacent cells is a factor that helps compensate for the spectral inefficiency of having to use processing gain to combat interference. A major advantage of CDMA in cellular telephony is the elimination of the need for frequency and timeslot coordination among cells.  $\square$

## PROBLEMS

- 18-1.** For the M12 framing format of Example 18-7, assume that synchronous multiplexing is used so that all the 481 information bits originate on the tributary streams at 1,544 kb/s. (In actuality pulse-stuffing synchronization is used, so this is a hypothetical problem.) For this assumption, determine the following:
- The output bit rate.
  - The time corresponding to one frame and superframe.
- 18-2.** Rework the parameters of the INTELSAT TDMA system of Example 18-10 assuming that the inputs are G.733 bit streams (Example 18-6). Assume each time-slot slot is assigned to a G.733 bit stream. Given the practical need for guard-times, etc., which primary stream, the G.732 or G.733, is likely to yield the largest number of 64 kb/s voiceband channels over the TDMA system?
- 18-3.** For the HDLC synchronization method described in Example 18-17, describe qualitatively what will happen or can happen when a single bit-error occurs in the following:
- The link frame start flag.
  - The link frame end flag.
  - The information packet.
- This will identify the situations which will be encountered and suggest recovery procedures that are required.
- 18-4.** Describe a minimum of two disciplines for the control unit in Figure 18-8 in addition to FIFO and roll-call polling.
- 18-5.**
- For the FIFO queueing discipline in a statistical multiplex, show that the average number of packets waiting in the multiplex buffers is  $\frac{\rho}{1 - \rho}$ .
  - Show that the probability that the buffer contains  $M$  or more packets is  $\epsilon$  for  $M = \frac{\log \epsilon}{\log \rho}$ .
  - What can you conclude about the size of a finite buffer required to maintain a certain probability of buffer overflow?
- 18-6.** Assume a statistical multiplex using a FIFO discipline has ten incoming links, each at 1 Mb/s, and one outgoing link at 2 Mb/s. Each incoming link has packets with exponentially distributed lengths, with an average length of 500 bits, and packets arriving on average every 3 msec.
- What is the utilization of each of the incoming links?
  - What is the utilization of the output link?
  - What is the average queueing delay through the multiplex?
- 18-7.** *Slotted Aloha* [15]. Show that the following simple modification of pure ALOHA results in a doubling of the throughput. Define time-slots with duration equal to the duration of one packet, and make these time-slots known to each node on the network. Each node then transmits its packets in alignment with the next time-slot after arrival of a packet.

## REFERENCES

1. M. Decina and A. Roveri, "ISDN: Architectures and Protocols," pp. 40 in *Advanced Digital Communications Systems and Signal Processing Techniques*, ed. K. Feher, Prentice-Hall, Englewood Cliffs, N.J. (1987).
2. G. H. Bennett, "Pulse Code Modulation and Digital Transmission," *Marconi Instruments*, (April 1978).
3. S. J. Campanella and D. Schaefer, "Time-Division Multiple Access Systems (TDMA)," *Prentice Hall*, (1983).
4. D. Reudink, "Advanced Concepts and Technologies for Communications Satellites," pp. 573 in *Advanced Digital Communications Systems and Signal Processing Techniques*, ed. K. Feher, Prentice-Hall, Englewood Cliffs, N.J. (1987).
5. D. E. Carlson, "Bit-Oriented Data Link Control Procedures," *IEEE Trans. on Communications* COM-28(4) p. 455 (April 1980).
6. M. Schwartz, *Telecommunication Networks: Protocols, Modeling, and Analysis*, Addison-Wesley, Reading, Mass. (1987).
7. M. R. Karim, "Delay-Throughput and Buffer Utilization Characteristics of Some Statistical Multiplexers," *AT&T Technical Journal* 66(March 1987).
8. N. Abramson, "The ALOHA System," *Prentice-Hall*, (1973).
9. J. I. Capetanakis, "The Multiple Access Broadcast Channel," *Ph.D. Thesis, Mass. Inst. Tech.*, (Aug. 1977).
10. J. L. Massey, "Channel Models for Random-Access Systems," *Proceedings NATO Advanced Study Institute*, (July 1986).
11. R.M. Metcalfe and D.R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications ACM* 19(7) p. 395 (July 1976).
12. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread-Spectrum Communications — A Tutorial," *IEEE Trans. Communications* COM-30(5) p. 855 (May 1982).
13. K. S. Gilhausen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver Jr, and C. E. Wheatley III, "On the Capacity of a Cellular CDMA System," *IEEE Trans. on Vehicular Technology* 40(May 1991).
14. R. J. Pickholtz, L. B. Milstein, and D. L. Shilling, "Spread Spectrum for Mobile Communications," *IEEE Trans. on Vehicular Technology* 40(May 1991).
15. L. G. Roberts, "ALOHA Packet System With and Without Slots and Capture," *Computer Communications Review* 5 p. 28 (April 1975).

# 19

---

## ECHO CANCELLATION

---

Multiple access communication, discussed in Chapter 18, usually relies on the orthogonality of the different signals by separating them in time or frequency. Analogous techniques apply to *full-duplex transmission*, or simultaneous transmission in both directions on a point-to-point link. Specifically, we can use *time-compression multiplexing (TCM)* and *frequency-division multiplexing (FDM)*. A more efficient approach, *echo cancellation* enables transmission in two directions simultaneously using the same frequency band, thereby reducing the bandwidth requirements approximately in half relative to TCM and FDM.

### Example 19-1.

Full-duplex digital transmission on a single wire pair from central office to telephone subscriber (the *digital subscriber loop*), as standardized in the United States, uses four-level baseband transmission. Given the communication engineer's penchant for obfuscation, this is called "2B1Q" line coding, which stands for "two bits on one quaternary digit". The bit rate is 160 kb/s, including 144 kb/s user data and 16 kb/s for framing and control, and the baud rate is therefore 80 kb/s. The bandwidth required on the cable is, for 0% excess bandwidth, 40 kHz. Both directions of transmission share this same bandwidth, with echo cancellation used to separate the two directions. See [1,2,3] for comparisons of the relative merits of echo cancellation and TCM in this application. □



**Example 19-2.**

The V.32 full-duplex modem transmits 9600 b/s in both directions over a voiceband data channel. It uses a baud rate of 2400 Hz, with four bits per symbol, and uses QAM modulation with a carrier frequency of 1800 Hz. With 0% excess bandwidth, the frequency band used would therefore be from 400 to 3000 Hz, nearly the full bandwidth of the voiceband data channel. TCM is unsuitable for voiceband data transmission because of the possibility of large propagation delays (such as on connections including a satellite link), and FDM is too bandwidth inefficient for higher speed modems. □

At each end of a full-duplex link, the near-end transmitted signal can be used to eliminate the undesired interference (called an *echo*) of the near-end transmitted signal at the receiver. An *echo canceler* can learn adaptively the response from near-end transmitter to receiver, generate a replica of that echo, and subtract that echo replica from the receiver input to yield an interference-free signal.

**Example 19-3.**

In principle, echo cancellation could be used to share any medium, such as a radio channel (Section 5.4), for the two directions. A radio channel would be of great practical interest because of the limited available spectrum, but unfortunately is impractical in today's technology because the speed, and particularly the accuracy, required for the echo cancellation. However, we cannot rule it out for the future. □

## 19.1. PRINCIPLE OF THE ECHO CANCELER

When we transmit full-duplex data, the primary problem is undesired feed-through of the transmitted data signal into the receiver through the hybrid. This extraneous signal is called *echo*. The operation of the hybrid was discussed in Section 5.5, and in particular it was illustrated in Figure 5-37, where the mechanism for echo was stated to be a mismatch between the impedance of the two-wire cable and the hybrid balancing impedance.

**Example 19-4.**

As shown in Figure 5-39, there are actually two opportunities for undesired echo on a voiceband data connection — the near-end hybrid and one or more far-end hybrids. One difficulty with the far-end echo that we will have to address is the possible frequency offset that it experiences, just as with the far-end data signal. The digital subscriber loop application is easier than the voiceband data canceler in this respect, since there is no far-end echo mechanism. □

The echo cancellation method of full-duplex transmission is illustrated in Figure 19-1. There is a transmitter (TR) and receiver (REC) on each end of the connection, and a hybrid is used to provide a virtual four-wire connection between the transmitter on each end and the receiver on the opposite end. The echo canceler is an adaptive transversal filter (Chapter 11) that adaptively learns the response of the hybrid, and generates a replica of that response which is subtracted from the hybrid output to yield an echo-free received signal.

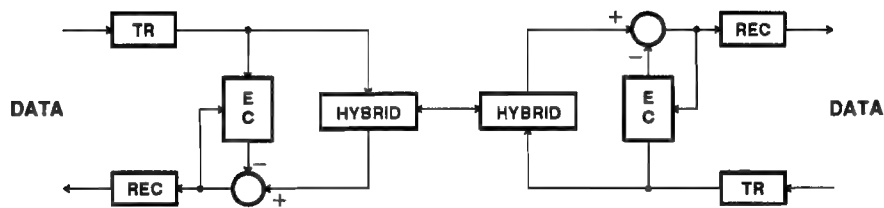


Figure 19-1. Echo cancellation method of full-duplex data transmission.

Example 19-5.

Typical numbers would be a 10 dB worst-case hybrid loss and 40 dB attenuation of the far-end transmitter. This implies a -30 dB signal-to-echo ratio at the hybrid output, which is clearly unacceptable. This can be improved to a more reasonable + 20 dB by an echo canceler with an additional 50 dB echo attenuation. □

The echo canceler notation is shown in Figure 19-2. The local transmitter signal  $y(t)$  at port A generates the undesired echo signal  $r(t)$ . This signal is superimposed at the output of the hybrid (port D) with the far transmitter signal  $x(t)$ . The canceler takes advantage of its knowledge of the local transmitter signal to generate a replica of the echo,  $\hat{r}(t)$ . This replica is subtracted from the echo plus far transmitter signal to yield  $e(t)$ , which ideally contains the far transmitter signal  $x(t)$  alone. The echo canceler is usually implemented in discrete-time as a finite transversal filter (Chapter 11) as shown in Figure 19-3. Essentially the same stochastic gradient algorithm can be applied to adapt the canceler to the details of the echo path response as was used to adapt the equalizer.

The canceler design depends strongly on the details of the local transmitter and receiver design.

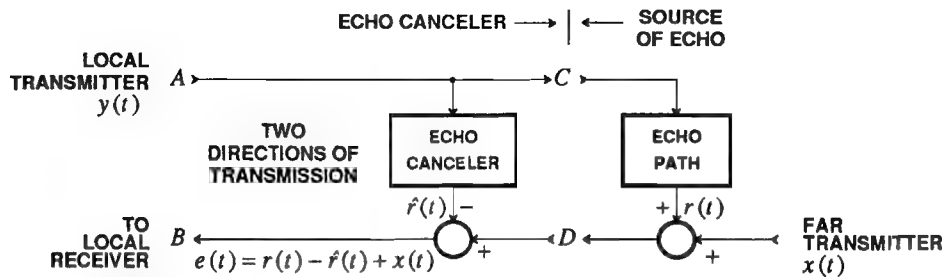


Figure 19-2. The principle and notation of an echo canceler.

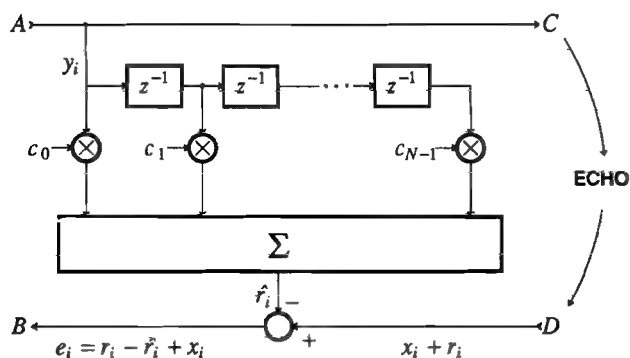


Figure 19-3. A transversal filter echo canceler.

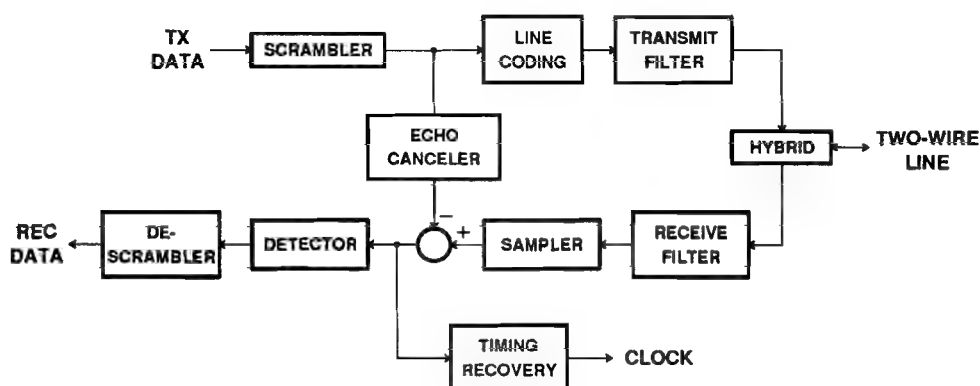


Figure 19-4. A block diagram of a full-duplex digital subscriber loop transceiver.

### Example 19-6.

A block diagram illustrating the functions of the digital subscriber loop transceiver is shown in Example 19-6 [4]. The transmit data is first scrambled (Section 12.5) to insure that there are sufficient pulses for timing recovery on the other end. Some form of line coding (Section 12.1) is applied to control the transmitted signal spectrum; for example, to insure that there is no energy at d.c. Next is the transmit filter to limit the high frequency components in the signal for radio-frequency interference (RFI) and crosstalk purposes. The echo canceler is connected either before or after the line coding at a point where the echo path is linear. A receive filter prevents aliasing in the subsequent sampling operation, and may also provide equalization of the high frequency attenuation of the cable. The signal is then sampled, since the echo canceler operates in the sampled data domain. After echo cancellation, the data is detected, taking into account the line coding and any intersymbol interference present, and descrambled to yield the received data sequence. The choice of sampling rate represents a tradeoff between the complexity of the echo canceler and the ease of recovering timing. For purposes of data detection, a sampling rate equal to the data

symbol rate is adequate, although there are many benefits to doubling this rate and using fractionally-spaced equalizers (Chapter 10). Timing recovery (Chapter 17) is usually considered to require a sampling rate equal to at least twice the data symbol rate (baud-rate timing recovery is also possible [5]). This implies that the echo canceler has different sampling rates at input and output, since the input sampling rate is equal to the baud rate. This is a major consideration in the echo canceler design, and is discussed in Section 19.2.  $\square$

## 19.2. BASEBAND CHANNEL

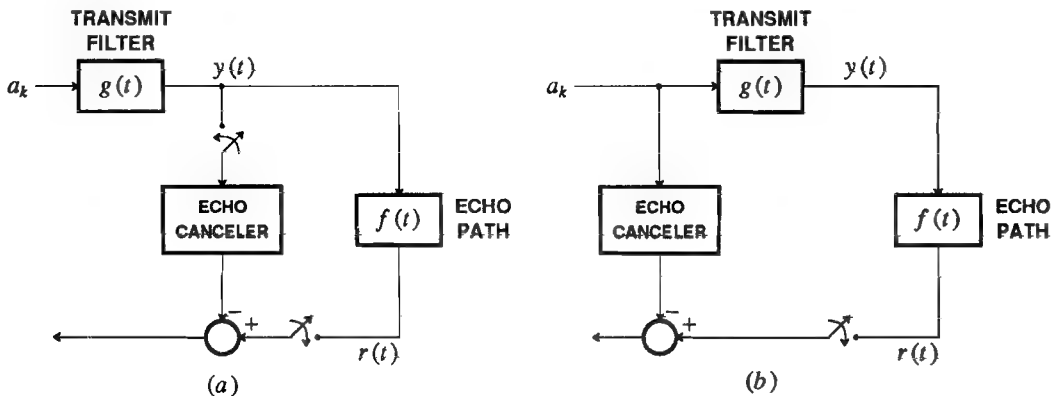
There are significant differences between the baseband and passband channel echo cancelers. We defer the more complicated passband canceler to Section 19.3. Assume a transmitted PAM signal is

$$y(t) = \sum_{m=-\infty}^{\infty} a_m g(t - mT), \quad (19.1)$$

where  $a_m$  is the sequence of transmitted data symbols,  $g(t)$  is the transmitted pulse shape,  $T$  is the baud interval, and the echo has transfer function  $F(j\omega)$ . Let  $h(t) = g(t) * f(t)$ , so the echo response is

$$r(t) = \sum_{m=-\infty}^{\infty} a_m h(t - mT). \quad (19.2)$$

Two approaches to echo cancellation are shown in Figure 19-5a. In Figure 19-5a we sample the transmitted data waveform  $y(t)$  at the canceler input, and in Figure 19-5b we apply the transmitted data symbols directly to the canceler so that the transmit filter is included in the echo path. Because the transmitted and echo signals will have



**Figure 19-5.** Two configurations for a baseband channel echo canceler. Cancellation using a. the sampled transmitted data waveform and b. the transmitted data symbols.

bandwidth greater than half the baud rate, a sampling rate of twice the baud rate or more will be required in Figure 19-5a. On the other hand, the sampling at the input of the canceler in Figure 19-5b is equal to the baud rate, leading to the immediate difficulty that the sampling rate at the output of the echo canceler is higher than the sampling rate at the input! This is precisely the opposite of the situation that we encountered in the fractionally-spaced equalizer in Chapter 10, where the input sampling rate was higher than the output.

### Interleaved Echo Cancelers

There is a ready solution to the problem of incompatible sampling rate, called *interleaved echo cancelers*. Since a clock representing the transmitted data signal is available, it is natural to sample the echo signal at a rate that is an integer multiple of the transmit baud rate, say a multiple  $R$ . Define a special notation for the samples of the received signal at this rate,

$$r_k(l) = r((k + \frac{l}{R})T), \quad 0 \leq l \leq R-1 \quad (19.3)$$

where the index  $k$  represents the data symbol epoch and  $l$  represents the sample from among  $R$  samples uniformly spaced in this epoch. This notation suggests an interpretation of this stream of samples as a set of  $R$  interleaved sample streams each with sampling rate equal to the baud rate. Similarly, define a notation for the samples of the echo pulse response

$$h_k(l) = h((k + \frac{l}{R})T), \quad 0 \leq l \leq R-1 \quad (19.4)$$

Combining the last three equations,

$$r_k(l) = \sum_{m=-\infty}^{\infty} h_m(l) a_{k-m} \quad (19.5)$$

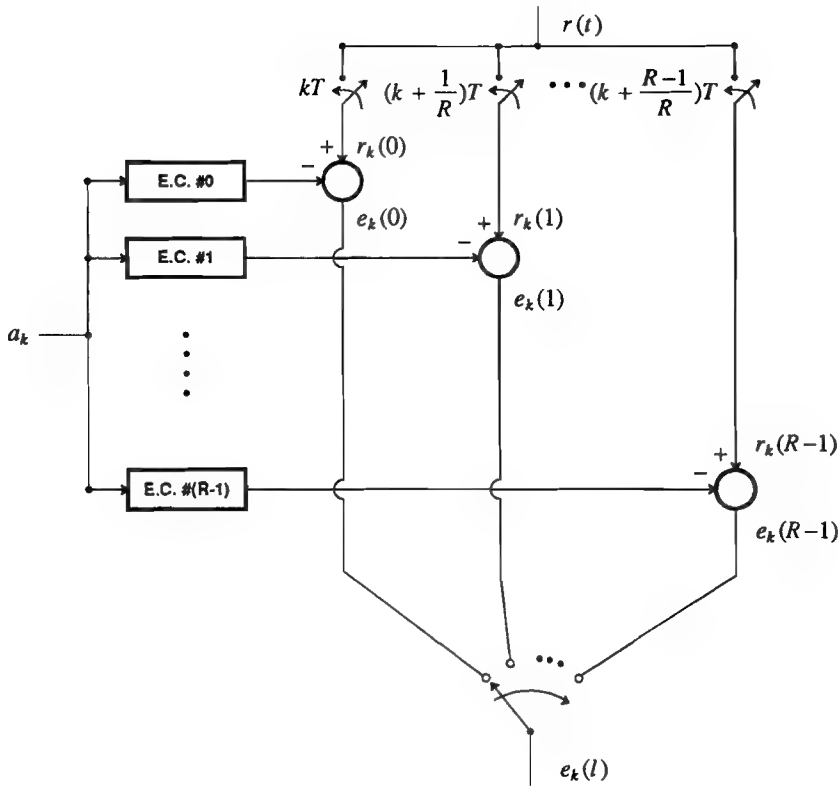
This relation shows that the samples of the echo can be thought of as  $R$  independent echo channels, each channel being driven by an identical sequence of data symbols. The discrete-time impulse response of the  $l$ -th echo channel is  $h_k(l)$ .

The echo replica can be generated independently for each echo channel by a set of  $R$  *interleaved echo cancelers* as shown in Figure 19-6. Each canceler cancels the echo for one sampling phase, from among  $R$ , and has a sampling rate at the input and output equal to the baud rate. Each canceler operates independently of the other; in particular, each generates its own error signal for purposes of both the full-duplex data receiver and the adaptation of the corresponding canceler.

Since the  $R$  echo channels are independent, the index  $l$  can be dropped. In the sequel we need only consider the design of one of the interleaved echo cancelers, and all the others follow naturally. The transversal filter echo canceler generates the replica

$$\hat{r}_k = \sum_{m=0}^{N-1} c_m a_{k-m} \quad (19.6)$$

where  $c_m$ ,  $0 \leq m < N$ , are the  $N$  filter coefficients of one of the  $R$  interleaved



**Figure 19-6.** A set of  $R$  interleaved echo cancelers, each canceling one of  $R$  phases.

transversal filters. This transversal filter generates an FIR approximation to the echo response  $h_m(l)$ .

Each canceler can be thought of as adapting to the impulse response of the echo channel sampled at a rate equal to the baud rate, but with a particular phase out of  $R$  possible phases. These cancelers independently converge, although they do have in common the same input sequence of data symbols. Since the transversal filters all adapt independently, the presence of multiple interleaved canceler filters does not affect the speed of adaptation. Therefore, the choice of an output sampling rate is purely a question of implementation complexity; the adaptation rate and asymptotic error are not affected by the sampling rate.

Returning to Figure 19-5, the interleaved canceler required in Figure 19-5b has important advantages over the configuration of Figure 19-5a [6]:

- The input to the canceler is transmitted data symbols, with a finite (and usually small) alphabet. The implementation of the canceler therefore requires a relatively simple multiplier, since the transmitted data symbols have a very few bits (perhaps as low as one) of precision.

- The speed of adaptation is greater, since the interleaved cancelers adapt independently and each has fewer taps.
- The canceler complexity as measured by the multiplication rate is lower, as illustrated by the following example.

**Example 19-7.**

If the sampling rate for the received signal is  $R$  samples per baud, and the effective length of the echo impulse response is  $N$  baud intervals (which we assume is not affected appreciably by the presence of the transmit filter in the echo response path), we can compare the multiplication rate for Figure 19-5a and b. In Figure 19-5a the convolution sum will have  $NR$  taps, each of which must be calculated  $R$  times per baud, for a total multiplication rate equal to  $NR^2$  times the baud rate. In Figure 19-5b each of the interleaved cancelers will have  $N$  taps, calculated at the baud rate, for a multiplication rate equal to  $N$  times the baud rate. Considering that there are  $R$  interleaved cancelers, the total multiplication rate is  $NR$  times the baud rate. The interleaved canceler therefore has a multiplication rate lower by a factor of  $R$ .  $\square$

For all these reasons, the configuration of Figure 19-5b is generally preferred over Figure 19-5a.

## 19.3. PASSBAND CHANNEL

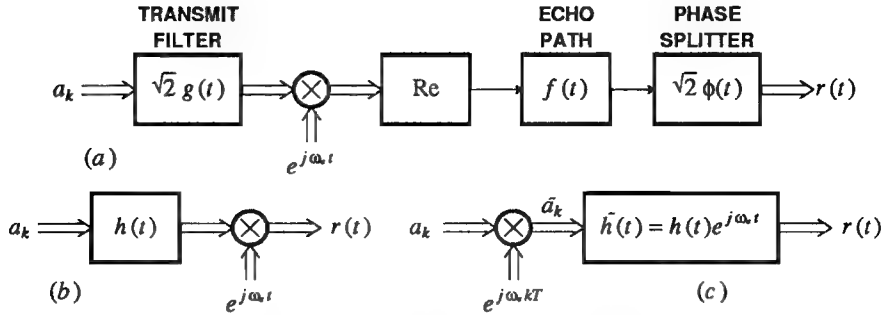
The *passband echo canceler* is considerably different from the baseband channel case. Assuming the data symbols are applied directly to the canceler as in Figure 19-5b, there are two obvious differences:

- The canceler input is complex-valued.
- The transmitter modulator is included in the transmit path, so that the echo path is *time-varying*. An adaptive filter could in principle model track this time varying channel, but in practice the required adaptation speed could not be achieved.

Fortunately, the carrier frequency and phase is precisely known, so that we can compensate for the carrier by adding a similar modulator to the transversal filter. There are numerous configurations possible, as we will see in this section. Pioneering work on the passband channel canceler was done by S. Weinstein of Bell Laboratories[7]. We begin by developing a model for the echo path, similar to the baseband model that was developed for the channel in Chapter 6.

### 19.3.1. Echo Path Model

A model for the transmitter and echo path is shown in Figure 19-7a. The transmit filter is  $g(t)$  and the echo path impulse response is  $f(t)$ . For the time being we assume that a phase splitter is included in the receive circuitry to generate the analytic signal, although we will see alternative configurations. The equivalent model shown in Figure 19-7b follows directly from the results of Chapter 6, since transmitting a passband PAM signal through a communication channel is no different from transmitting through an echo channel. From (6.57), the received echo signal is



**Figure 19-7.** The transmitter and echo path for a passband channel echo canceler. a. The transmitter, echo path, and a phase splitter at the receiver input. b. An equivalent model for the path from transmitted data symbols to received analytic signal consisting of a baseband echo channel followed by modulator. c. An alternative model consisting of a modulator followed by passband echo channel.

$$r(t) = \sqrt{2} \operatorname{Re} \left\{ \sum_{k=-\infty}^{\infty} a_k h(t - kT) e^{j\omega_c t} \right\}, \quad (19.7)$$

where the equivalent baseband complex-valued response is, from (6.58),

$$h(t) = (f(t) e^{-j\omega_c t}) * g(t), \quad H(j\omega) = F(j(\omega + \omega_c)) G(j\omega). \quad (19.8)$$

The conclusion is that the echo channel output can be considered as a signal of the same form as the transmitted signal, except the transmitted baseband pulse  $g(t)$  has been replaced by an echo-channel equivalent baseband output pulse  $h(t)$ . The latter is obtained by shifting the echo transfer function in the vicinity of the carrier frequency down to d.c. Since  $h(t)$  is in general complex-valued, even though the transmit pulse  $g(t)$  is real-valued, the echo canceler must have complex-valued tap coefficients! This of course implies that there is crosstalk between the in-phase and quadrature channels when they pass through the echo channel, similarly to the situation in channel equalization.

After a minor manipulation, the analytic signal corresponding to (19.7) at the output of a phase splitter can be written in the form

$$r(t) = \sum_k a_k e^{j\omega_c kT} h(t - kT) e^{j\omega_c (t - kT)} = \sum_k \tilde{a}_k \tilde{h}(t - kT) \quad (19.9)$$

where

$$\tilde{h}(t) = h(t) e^{j\omega_c t} \quad (19.10)$$

is an equivalent passband pulse waveform and

$$\tilde{a}_k = a_k e^{j\omega_c kT} \quad (19.11)$$

is called the *rotated data symbol* since it is simply rotated by angle  $\omega_c kT$  radians. This results in the model of Figure 19-7c. The rotation of the data symbols is in effect a modulation up to passband, and then the rotated symbols are put through an



equivalent passband channel with impulse response  $\tilde{h}(t)$ . Since  $h(t)$  is a baseband pulse, this filter has a response centered at the carrier frequency. The rotation of the data symbols is simple to implement when the carrier frequency and baud rate have a simple relationship.

#### Example 19-8.

For a V.32 modem, the carrier frequency is 1800 Hz and the baud rate is 2400 Hz. Therefore,

$$\omega_c T = 2\pi \times 1800 \times \frac{1}{2400} = \frac{3\pi}{2} \text{ radians} . \quad (19.12)$$

For this case, the exponent  $\omega_c kT$  assumes only multiples of  $\pi/2$ , and hence the rotation requires only multiplication by values that are of the form  $\pm 1$  or  $\pm j$ . The rotation in this case is always by some multiple of 90 degrees.  $\square$

### 19.3.2. Interleaved Passband Channel Echo Cancelers

Just as in the baseband case, the sampling rate at the receiver input will generally be a multiple  $R$  of the baud rate, necessitating interleaved echo cancelers. Defining  $r_i(l)$ ,  $h_i(l)$ , and  $\tilde{h}_i(l)$  as in (19.3) and (19.4), a relation similar to (19.5) is obtained. For the echo channel model of Figure 19-7b, we get

$$r_k(l) = \left[ \sum_m a_m h_{k-m}(l) \right] e^{j\omega_c(k + \frac{l}{R})T} . \quad (19.13)$$

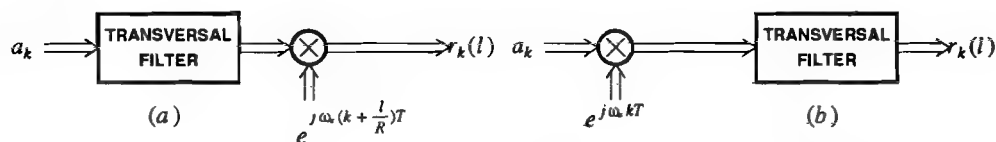
and for the echo channel model of Figure 19-7c we get

$$r_k(l) = \sum_m \tilde{a}_m \tilde{h}_{k-m}(l) . \quad (19.14)$$

In both cases we can implement the canceler as  $R$  independent interleaved cancelers.

### 19.3.3. Passband vs. Baseband Transversal Filters

Based on the discrete-time interleaved representations for the echo channel represented by (19.13) and (19.14), there are two echo canceler configurations to synthesize these echo responses as pictured in Figure 19-8. The difference between these two configurations is the placement of the modulator after or before the complex-coefficient transversal filter.



**Figure 19-8.** Two configurations for one interleaved echo canceler corresponding to a passband channel. a. A baseband transversal filter followed by modulator. b. A modulator followed by a passband transversal filter.

The baseband transversal filter of Figure 19-8a follows directly from the representation of Figure 19-7b and (19.13). Let the transversal filter have  $N$  complex-valued coefficients  $c_k$ ,  $0 \leq k \leq N-1$ , in which case the echo canceler can be represented mathematically as

$$\hat{r}_k(l) = \left[ \sum_{m=0}^{N-1} c_m a_{k-m} \right] e^{j\omega_c(k + \frac{l}{R})T}. \quad (19.15)$$

This can be represented as a transversal filter, which performs the convolution sum, followed by a modulator. The transversal filter is approximating the equivalent baseband pulse  $h(t)$  in the model of Figure 19-7b, and hence we call it a *baseband transversal filter*.

An equivalent configuration follows from the model of Figure 19-7c and (19.14), from which we get an echo canceler of the form

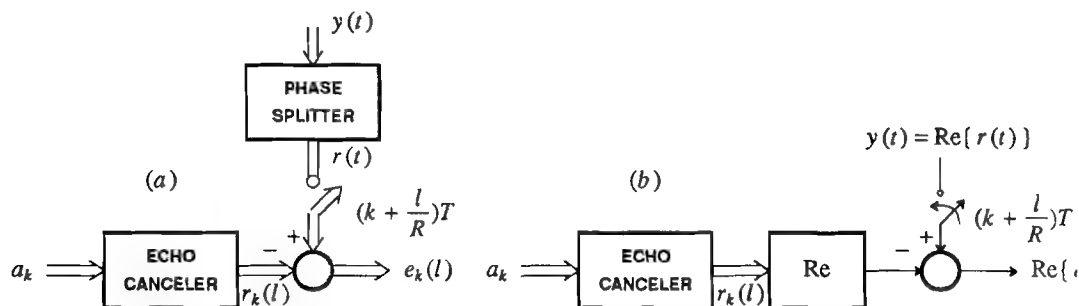
$$\hat{r}_k(l) = \sum_{m=0}^{N-1} c_m \tilde{a}_{k-m}. \quad (19.16)$$

This configuration is shown in Figure 19-8b. The rotator first modulates the data symbols to passband, and the transversal filter then approximates the passband response  $\tilde{h}(t) = h(t)e^{j\omega_c t}$ . For this reason we call this a *passband transversal filter*.

### 19.3.4. Real vs. Complex Error Cancelers

Yet another option in cancelers for a passband channel is the generation of a real-valued or complex-valued error signal as illustrated in Figure 19-9. The complex-error configuration of Figure 19-9a is the one considered thus far in this section. It is assumed that the receive analytic signal is generated using a phase splitter, and the echo canceler generates a replica of the echo analytic signal.

The real-error alternative shown in Figure 19-9b cancels only the real part of the analytic signal, which in actuality is the passband receive waveform. For this case,



**Figure 19-9.** Two options for passband channel echo cancellation (shown is one of  $R$  interleaved cancelers). a. Complex-error canceler, requiring phase splitter before cancellation. b. Real-error canceler, requiring no phase splitter.

only the real part of the canceler complex-valued output is required. Use of the real-error canceler can reduce the canceler computational load because only the real part of the output need be calculated. Similarly, the receive signal is used in place of the analytic signal, the former being the real part of the latter, thereby eliminating the need for the phase splitter. Overall, then, the complexity of the real-error canceler is lower.

We will see in the next section that the convergence of the complex-error canceler is faster than that of the real-error canceler, because the former makes use of more information. Further, in some circumstances the savings of a phase splitter in a real-error canceler is negated by the need for a splitter in the data receiver that follows the echo canceler. On the other hand, the real-error canceler is especially attractive in the Nyquist cancellation application described in the next subsection.

### 19.3.5. Nyquist Cancellation

In voiceband data modems, the two directions of transmission are governed by independent clocks, and therefore there will be a frequency offset. This implies that the sampling clock used for echo cancellation is not necessarily the same in frequency or phase as the appropriate sampling clock for recovery of the far-end data. The usual solution to this problem is to use a *Nyquist canceler* which operates at a sufficiently high sampling rate to allow recovery of a continuous-time version of the far-end data signal. This far-end signal can then be resampled in accordance with the appropriate clock without regard to its phase or frequency relative to the transmit data clock.

The Nyquist canceler is shown in Figure 19-10. The canceler works on samples generated synchronously with the transmit data stream (the dashed line indicates the source of the clock for each sampler). The bandpass filter (BPF) prior to the sampler eliminates all noise out-of-band of the received data signal (and incidently some of the echo as well). The sampling rate is chosen to be Nyquist; that is, greater than twice the highest frequency in the receive data signal. For convenience it will be an integral multiple of the transmit baud rate clock.

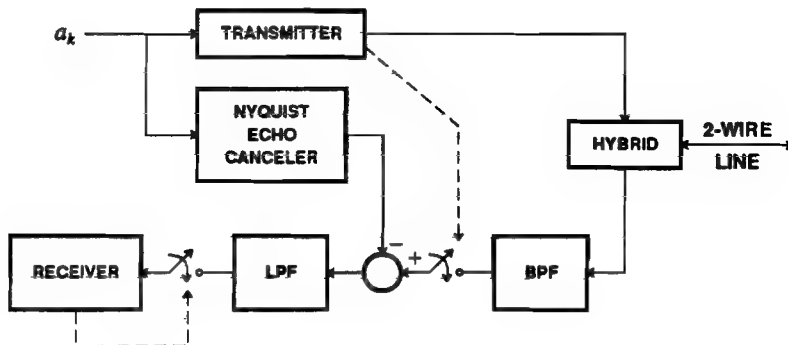


Figure 19-10. A Nyquist echo canceler operating synchronously with the transmitted data stream and asynchronously from the data receiver.

After recovery of the receive data signal without the echo, a continuous-time receive signal is recovered using a lowpass filter (LPF). This signal is then resampled synchronously with the receive data signal using a clock provided by the receiver.

An advantage of the Nyquist canceler is that an existing half-duplex data receiver can be used without modification. The purpose of the echo cancellation "front-end" is merely to eliminate the undesired echo interference from the transmitter. Note also that a real-error canceler has been used and is attractive in this configuration, since there is a savings of a phase splitter. Only one phase splitter is required, the one in the receiver.

## 19.4. ADAPTATION

As with adaptive equalizers (Chapter 11), there are two measures of performance of an adaptive echo canceler: the speed of adaptation and the accuracy of the cancellation after adaptation. There is a tradeoff between these two measures: for a particular class of adaptation algorithm, as the speed of adaptation is increased the accuracy of the transfer function after adaptation gets poorer. This tradeoff is fundamental, since a longer averaging time is necessary to increase asymptotic accuracy, but slows the rate of convergence. Usually the motivation for adapting an echo canceler is that the transfer function of the echo is not known in advance. It is also probable that the echo transfer function is changing with time, although in most cases the change will be quite slow (say in response to changes in the temperature of the transmission facilities). Thus, in most instances the accuracy of the final cancellation of the echo is the most critical design factor.

### Example 19-9.

In the digital subscriber loop, the transceiver will often be dedicated to a particular loop. As long as the transceiver is allowed to run all the time, or at least stores the echo canceler coefficients between calls, the adaptation can be quite slow (resulting in a high accuracy) because the echo path transfer function should change only in response to temperative changes and similar influences, which occur quite slowly. □

Although the ability of the canceler to rapidly track a changing echo response is usually not important, the speed of initial adaptation from an arbitrary initial condition is often important.

### Example 19-10.

In a voiceband data modem, the echo canceler must converge anew at the beginning of each call. The adaptation of the echo canceler is therefore a part of the initialization sequence before useful data transmission can occur. Since one would like to minimize that initialization time, there is motivation to adapt as quickly as possible. This is a natural application for a gear-shifting algorithm, since the accuracy of cancellation is not critical during the training period (no actual data transmission is taking place) and therefore it is permissible to start with a larger step-size. With respect to the the far-end echo canceler (Section 19.6) a more rapid tracking capability will be required. □

We will derive a SG adaptation algorithm for the complex-error passband transversal filter algorithm in this section[8]. The case of a baseband channel canceler is a special case [9,10,11] and will also be covered. The adaptation of the baseband transversal filter canceler for the passband channel is a simple extension and is relegated to the problems [7]. The adaptation of the real-error canceler is a bit more complicated to derive and analyze and is relegated to appendix 18-A. More general results on adaptation algorithms and their convergence can be found in [12].

As usual, we consider the minimum MSE problem first, followed by the SG algorithm. In all cases we will derive the adaptation algorithm for only one of the  $R$  interleaved cancelers, and assume that same algorithm is applied identically to all.

### 19.4.1. Minimum MSE Solution

In this section we consider the optimum tap coefficients for a complex-error passband transversal filter canceler. Write the  $m$ -th filter coefficient as  $c_m$  and the analytic echo cancellation error at time  $k$  as  $E_k$ . Define a notation for the vector of  $N$  filter coefficients

$$\mathbf{c}' \equiv [c_0, c_1, \dots, c_{N-1}]. \quad (19.17)$$

For the passband transversal filter canceler, the input to the transversal filter is the rotated data symbol  $\tilde{A}_k$ . Define a vector of the current and  $N-1$  past input rotated data symbols

$$\tilde{\mathbf{a}}_k' \equiv [\tilde{A}_k, \tilde{A}_{k-1}, \dots, \tilde{A}_{k-N+1}]. \quad (19.18)$$

If the impulse response of the echo channel is  $\tilde{h}_k(l)$ ,  $0 \leq k < \infty$  for the  $l$ -th interleaved canceler, then it is also convenient to define a vector of the first  $N$  of these impulse response samples,

$$\tilde{\mathbf{h}}' = [\tilde{h}_0, \tilde{h}_1, \dots, \tilde{h}_{N-1}] \quad (19.19)$$

where in this and subsequent equations the " $l$ " is suppressed. All of these quantities are complex-valued, except in the baseband channel case where they are real-valued.

With this notation in hand, the analytic error signal can be written as

$$\begin{aligned} E_k &= \sum_{m=0}^{\infty} \tilde{h}_m \tilde{A}_{k-m} - \sum_{m=0}^{N-1} c_m \tilde{A}_{k-m} + X_k \\ &= (\tilde{\mathbf{h}} - \mathbf{c})' \tilde{\mathbf{a}}_k + V_k \end{aligned} \quad (19.20)$$

where  $X_k$  is the far-end data signal plus noise and

$$V_k = \sum_{m=N}^{\infty} \tilde{h}_m \tilde{A}_{k-m} + X_k \quad (19.21)$$

is the residual uncanceled echo. This uncanceled echo has several components:

- Echo components with delays that exceed the number of coefficients in the transversal filter,
- The noise introduced on the channel from the far-end data transmitter, and

- The far-end data signal, which represents a noise with respect to the adaptation of the echo canceler.

For the MSE solution, we assume that  $\tilde{A}_k$  is a wide-sense stationary discrete-time random process and that the echo channel  $h_k$  is known. We want to minimize the MSE error  $E(|E_k|^2)$ . This error signal includes, as one component, the far-end data signal, which we don't wish to minimize. Fortunately, the echo canceler has no influence over this data signal. As long as the data signals in the two directions are uncorrelated, minimizing the MSE will be the same as minimizing the component of echo in the error signal (as we will see).

Following consistent notation to Section 11.2, define

$$\mathbf{p} = E[V_k \tilde{\mathbf{a}}_k], \quad \Phi = E[\tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k'], \quad (19.22)$$

where these quantities are independent of  $k$  due to the wide-sense stationarity assumption. The autocorrelation of the rotated data symbols is easily related to the autocorrelation of the data symbols itself.

#### Exercise 19-1.

Show that the relationship between the rotated and non-rotated data symbol autocorrelation functions is,

$$E[\tilde{A}_k \tilde{A}_m^*] = e^{j\omega_c(k-m)T} E[A_k A_m^*]. \quad (19.23)$$

This demonstrates that if the data symbols are wide-sense stationary, then so too are the rotated symbols, where the relationship between the power spectra is

$$S_{\tilde{a}}(e^{j\omega T}) = S_a(e^{j(\omega-\omega_c)T}). \quad (19.24)$$

□

A simplification of the analysis of the echo canceler relative to the adaptive equalizer is that we can generally assume that the successive input data symbols are uncorrelated. This implies that the power spectrum is white, and from (19.24) the rotated data symbols are white also. In addition, since  $|\tilde{A}_k| = |A_k|$  the rotated symbols have the same variance as the symbols themselves. It follows for this case that the autocorrelation matrix  $\Phi$  is diagonal,

$$\Phi = \sigma_a^2 \mathbf{I}, \quad \sigma_a^2 = E[|A_k|^2]. \quad (19.25)$$

Explicitly evaluating the mean-square error,

$$E[|E_k|^2] = (\tilde{\mathbf{h}} - \mathbf{c})^* \Phi (\tilde{\mathbf{h}} - \mathbf{c}) - 2\text{Re}\{(\tilde{\mathbf{h}} - \mathbf{c})^* \mathbf{p}\} + \sigma_v^2, \quad (19.26)$$

where  $\sigma_v^2 = E[|V_k|^2]$  is the variance of the uncanceled echo. This is a Hermitian form in the tap weight vector  $\mathbf{c}$ , and hence there is a unique minimum. (19.26) can be written in the form

$$E[|E_k|^2] = \xi_{\min} + (\mathbf{c} - \mathbf{c}_{\text{opt}})^* \Phi (\mathbf{c} - \mathbf{c}_{\text{opt}}) \quad (19.27)$$

where

$$\mathbf{c}_{\text{opt}} = \tilde{\mathbf{h}} + \Phi^{-1} \mathbf{p}, \quad \xi_{\min} = \sigma_v^2 - \mathbf{p}^* \Phi^{-1} \mathbf{p}. \quad (19.28)$$

**Example 19-11.**

For the autocorrelation of (19.25), this solution reduces to

$$\mathbf{c}_{\text{opt}} = \tilde{\mathbf{h}} + \frac{1}{\sigma_a^2} \mathbf{p}, \quad \xi_{\min} = \sigma_v^2 - \frac{1}{\sigma_a^2} \|\mathbf{p}\|^2. \quad (19.29)$$

□

The  $\Phi$  matrix is Hermitian and non-negative definite, and has non-negative real-valued eigenvalues.

For the optimal solution to be unique, we have to assume  $\Phi$  is positive-definite, implying it is invertible, in which case this inverse  $\Phi^{-1}$  is also a Hermitian matrix. In this event, the second term in (19.27) is non-negative and has a unique minimum  $\mathbf{c} = \mathbf{c}_{\text{opt}}$ . This choice also minimizes the mean-square error, with resultant minimum value  $E[|E_k|^2] = \xi_{\min}$ .

**Example 19-12.**

If the data symbols are uncorrelated with the uncanceled error, or  $\mathbf{p} = \mathbf{0}$ , then the optimum tap weight vector is equal to the echo impulse response  $\mathbf{c}_{\text{opt}} = \tilde{\mathbf{h}}$  and the resultant mean-square error is equal to the variance of the uncanceled echo,  $\xi_{\min} = \sigma_v^2$ . The optimum coefficient vector and resulting MSE are independent of the autocorrelation matrix  $\Phi$ . This condition will hold when the far-end data signal  $X_k$  is uncorrelated with the near-end data symbols. When that condition is violated, the optimum coefficient vector is not equal to the echo impulse response. This imposes a system requirement for proper operation that the data symbols in the two directions be uncorrelated. If this is violated, the echo cancellation adaptation will be biased away from replicating the echo impulse response. □

## 19.4.2. Stochastic Gradient (SG) Algorithm

As with adaptive equalization, the most widely used adaptation algorithm for the echo canceler is the stochastic gradient (SG) algorithm. This is very similar to the algorithm we derived for adaptive equalizers in Chapter 11.

Consider the passband transversal filter case. The first step is to determine the magnitude-squared of the analytic cancellation error as a function of the coefficient vector  $\mathbf{c}$ ,

$$|E_k|^2 = |R_k - \mathbf{c}' \tilde{\mathbf{a}}_k|^2 = |R_k|^2 - 2\text{Re}\{\mathbf{c}^* R_k \tilde{\mathbf{a}}_k^*\} + \mathbf{c}^* \tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k' \mathbf{c} \quad (19.30)$$

and then we take the gradient of this expression with respect to  $\mathbf{c}$ . In view of Exercise 11-5 and the fact that the matrix  $\tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k'$  is Hermitian, we get

$$\nabla_{\mathbf{c}} |E_k|^2 = 2\tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k' \mathbf{c} - 2R_k \tilde{\mathbf{a}}_k^* = -2E_k \tilde{\mathbf{a}}_k^*. \quad (19.31)$$

The SG algorithm follows from evaluating this gradient at the last coefficient vector, multiplying by step-size  $\beta/2$ , and subtracting the result from the last coefficient vector to get the new coefficient vector,

$$\mathbf{c}_k = \mathbf{c}_{k-1} + \beta E_k \tilde{\mathbf{a}}_k^* \quad (19.32)$$

$$E_k = R_k - \mathbf{c}_{k-1}' \tilde{\mathbf{a}}_k. \quad (19.33)$$

The implementation of this algorithm is very similar to the adaptive equalizer case of Chapter 11, with one important difference; namely, the input samples  $\tilde{A}_k$  are the rotated transmit data symbols, and are typically drawn from a relatively small alphabet. This can simplify the implementation of the multiplications in both the convolution sum and the adaptation algorithm. The baseband channel case follows as a special case, where all quantities are real-valued and  $\omega_c = 0$ . The derivation of SG adaptation algorithms for other canceler structures of interest is relegated to exercises.

#### Exercise 19-2.

Show that the stochastic gradient adaptation algorithm for the complex-error canceler with baseband transversal filter is

$$\mathbf{c}_k = \mathbf{c}_{k-1} + \beta e^{j\omega_c(k + \frac{l}{R})T} E_k \mathbf{a}_k^* \quad (19.34)$$

$$E_k = R_k - e^{j\omega_c(k + \frac{l}{R})T} \mathbf{c}_{k-1}' \mathbf{a}_k. \quad (19.35)$$

**HINT:** See the hint for Problem 19-6.  $\square$

#### Exercise 19-3.

Show that the stochastic gradient adaptation algorithm for the real-error canceler with passband transversal filter is

$$\mathbf{c}_k = \mathbf{c}_{k-1} + \beta \operatorname{Re}\{E_k\} \mathbf{a}_k^* \quad (19.36)$$

$$\operatorname{Re}\{E_k\} = \operatorname{Re}\{R_k\} - \operatorname{Re}\{\mathbf{c}_{k-1}' \mathbf{a}_k\}. \quad (19.37)$$

$\square$

In the remainder of this section we will consider the convergence properties of the adaptation algorithm. Since the convergence analysis is so similar to the adaptive equalization case of Chapter 11, we can draw many results from there.

### 19.4.3. Convergence of the SG Algorithm

Defining a coefficient error vector

$$\mathbf{q}_k = \mathbf{c}_k - \mathbf{c}_{\text{opt}}, \quad (19.38)$$

the first step is to derive a stochastic difference equation for this error vector.

#### Exercise 19-4.

Define a stochastic matrix

$$\Gamma_k = \mathbf{I} - \beta \tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k', \quad (19.39)$$

and define the error of the optimal fixed coefficient echo canceler,



$$D_k = R_k - c_{\text{opt}}' \hat{\mathbf{a}}_k. \quad (19.40)$$

Then show that the coefficient error vector is governed by the stochastic difference equation

$$\mathbf{q}_k = \Gamma_k \mathbf{q}_{k-1} + \beta D_k \mathbf{a}_k^*. \quad (19.41)$$

□

We can also determine the excess MSE of the canceler directly from (19.27),

$$E[|E_k|^2] = \xi_{\min} + E[\mathbf{q}_k^{*'} \Phi \mathbf{q}_k]. \quad (19.42)$$

These results are identical to the stochastic difference equation derived for the adaptive equalizer in (11.60), with minor changes in notation reflecting the different application, and so we can use the results derived there directly in analyzing the echo canceler.

In particular, for the case of uncorrelated data symbols of (19.25), (19.42) becomes

$$E[|E_k|^2] = \xi_{\min} + \sigma_a^2 E[\|\mathbf{q}_k\|^2] \quad (19.43)$$

where the expected vector norm approximately obeys the difference equation

$$E[\|\mathbf{q}_{k+1}\|^2] = \gamma E[\|\mathbf{q}_k\|^2] + \beta^2 N \sigma_a^2 \xi_{\min}. \quad (19.44)$$

$$\gamma = 1 - 2\beta\sigma_a^2 + \beta^2 N \sigma_a^4. \quad (19.45)$$

The time constant of convergence of MSE can be obtained from setting  $\gamma^T = 1/e$ , from which we get

$$\tau \approx \frac{1}{2\beta\sigma_a^2}. \quad (19.46)$$

The maximum convergence rate for excess MSE is reached at

$$\beta_{\text{opt}} = \frac{1}{N\sigma_a^2} \quad (19.47)$$

with a resulting time constant

$$\tau \approx \frac{N}{2}. \quad (19.48)$$

The asymptotic excess MSE from (19.44) is

$$E[\|\mathbf{q}_k\|^2] \rightarrow \frac{N\beta}{2 - N\beta\sigma_a^2} \xi_{\min}. \quad (19.49)$$

and at the optimum step-size (optimum in terms of rate of convergence of MSE, not the asymptotic MSE), the asymptotic error is

$$E[\|\mathbf{q}_k\|^2] \rightarrow \frac{1}{\sigma_a^2} \xi_{\min}. \quad (19.50)$$

In view of (11.66), the asymptotic MSE is

$$E[|E_k|^2] \rightarrow \xi_{\min} + \xi_{\min} = 2\xi_{\min}. \quad (19.51)$$

Thus, for the fastest convergence, the total MSE is twice the minimum MSE for a fixed coefficient filter, with half that MSE attributable to the asymptotic wandering of the filter coefficients about their optimum value.

### Example 19-13.

Continuing Example 19-12, since  $\xi_{\min}$  is the variance of the uncanceled error for this case, the asymptotic MSE is for this case

$$E[|E_k|^2] \rightarrow 2\sigma_v^2. \quad (19.52)$$

Since (hopefully) the dominant component of the uncanceled error is the far-end data signal, this implies that for the choice of the optimum step-size the SNR, defined as the ratio of the far-end data signal power to excess MSE for cancellation, will be 0 dB. In words, the residual echo will have the same power as the received signal. This is, of course, not practical, so a smaller step-size resulting in slower convergence will be required.  $\square$

The analysis of convergence applies equally well to the baseband channel case, virtually without modification. The baseband transversal filter canceler analysis is also straightforward based on the results so far (Problem 19-7). The real-error canceler is a bit more complicated, and hence is relegated to Appendix 19-A. The results there can be summarized succinctly as follows. For the same step-size, the real-error canceler converges with a time constant that is approximately twice as great as the complex-error canceler. In retrospect, this is not surprising since the real-error canceler is in effect throwing away half the information available (the imaginary part of the analytic error). Both cancelers have approximately the same asymptotic MSE. Thus, we must trade off the (in some circumstances) simpler implementation of the real-error canceler against its poorer convergence properties.

## 19.5. FAR-END ECHO

In the voiceband data modem, echo can occur not only at the near-end in conjunction with the four-wire to two-wire converter, but also at intermediate points in the telephone network. These echos are generally more attenuated than the near-end echo, and hence require a less accurate cancellation, but they are also subject to additional impairments such as jitter and frequency offset. Hence, very accurate cancellation of these echos requires the addition of algorithms to the basic echo canceler considered thus far.

As always, the structure of the echo canceler depends on the assumed model for the far-end echo mechanism. One such model is shown in Figure 19-11. We have added to the usual passband filter operating on the rotated symbols two additional features:

- A *bulk delay* accounting for the propagation delay from the transmitter to the point of echo generation.

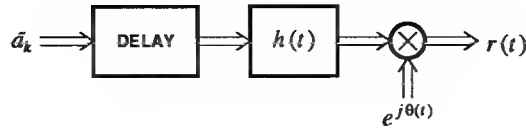


Figure 19-11. Model of far-end echo generation mechanism.

- A carrier phase rotation by angle  $\theta(t)$  at the output to account for possible phase jitter and frequency offset in the echo channel (frequency offset would of course result in a linearly increasing phase component).

A possible configuration for a voiceband data modem echo canceler based on this model is shown in Figure 19-12. We have shown the passband transversal filter with complex-error for convenience. The near-end echo canceler is identical to that considered earlier in this chapter — we do not expect to experience phase jitter or frequency offset in this echo path since the primary source of this echo is the hybrid within the voiceband data modem itself. The far-end echo canceler, however, replicates the model of Figure 19-11. It consists of a bulk delay, which hopefully matches the delay of the echo channel, a passband transversal filter, and a phase rotator by angle  $\hat{\theta}_k$  which hopefully matches the carrier phase rotation  $\theta_k$  of the echo channel. The appropriate angle for rotation is determined by a phase-locked loop, which uses the transversal filter output and cancellation error to correct the currently used phase in a similar manner to the carrier recovery circuitry discussed in Chapter 16.

The model of Figure 19-11 and hence the structure of Figure 19-12 may be oversimplified. For example, the actual channel may have filtering before and after the phase rotation, rather than just before as shown in Figure 19-11. Such a situation will require a correspondingly more complicated echo canceler structure.

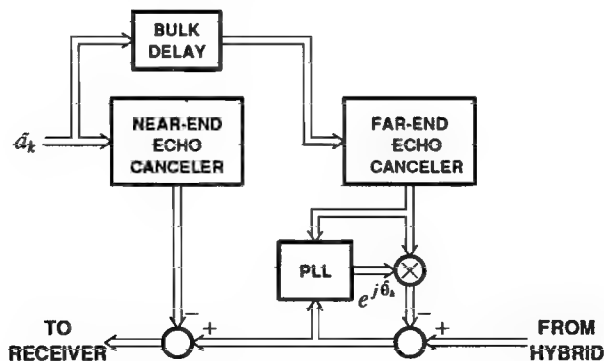


Figure 19-12. Passband echo canceler for a voiceband data modem with complex cancellation.

A PLL algorithm can be derived using a stochastic gradient (SG) approach, in which we take the derivative of  $|E_k|^2$  with respect to the PLL output phase  $\hat{\theta}_k$ .

### Exercise 19-5.

Show that for error

$$E_k = R_k - e^{j\hat{\theta}_k} \mathbf{c}_k' \tilde{\mathbf{a}}_k \quad (19.53)$$

the derivative of the MSE with respect to  $\hat{\theta}$  is

$$\frac{\partial |E_k|^2}{\partial \hat{\theta}} = -2 \text{Im} \{ e^{-j\hat{\theta}} E_k \mathbf{c}_k' \tilde{\mathbf{a}}_k \} . \quad (19.54)$$

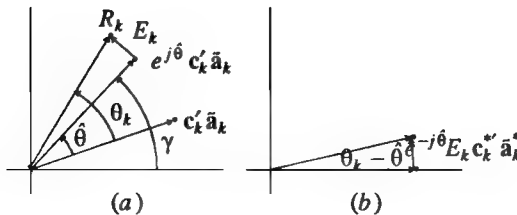
□

By adjusting  $\hat{\theta}_k$  in the opposite direction of this derivative, we can track the phase error. The result in (19.54) has a simple and intuitive interpretation shown in Figure 19-13. As shown in Figure 19-13a, the goal is for the echo replica  $e^{j\hat{\theta}} \mathbf{c}_k' \tilde{\mathbf{a}}_k$  to equal the echo signal  $R_k$ , or to have the error signal  $E_k$  equal to zero. The diagram assumes that there is no far-end data signal or noise ( $R_k$  consists only of echo) and that the echo canceler transversal filter has converged so that the only difference between  $R_k$  and the echo replica is a phase rotation by  $\theta_k - \hat{\theta}$ . Under these assumptions, the actual echo is  $\mathbf{c}_k' \tilde{\mathbf{a}}_k$  rotated by  $\theta_k$ , and the echo replica is  $\mathbf{c}_k' \tilde{\mathbf{a}}_k$  rotated by  $\hat{\theta}$ .

Now, multiplying  $E_k$  by  $e^{-j\hat{\theta}} \mathbf{c}_k'' \tilde{\mathbf{a}}_k^*$  is equivalent to rotating the entire constellation by  $-\gamma$ , where  $\gamma$  is the angle of the echo replica relative to the real-axis. This rotation places the echo replica on the real-axis as shown in Figure 19-13b. In this rotated constellation it is easy to tell whether the phase error  $\theta_k - \hat{\theta}$  is positive or negative by examining the imaginary-part of the rotated error. If this imaginary-part is positive, the error is reduced by making the estimated phase  $\hat{\theta}$  larger.

A SG PLL algorithm for adjustment of the phase follows from the derivative in Exercise 19-5,

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \beta \text{Im} \{ E_k e^{-j\hat{\theta}} \mathbf{c}_k'' \tilde{\mathbf{a}}_k^* \} . \quad (19.55)$$



**Figure 19-13.** Interpretation of far-end echo canceler phase detector. a. Transversal filter and echo channel output. b. Rotated so that the echo replica is on the real-axis.

**Exercise 19-6.**

Show that if the echo canceler transversal filter has converged and there is no far-end signal or noise,

$$E[\operatorname{Im}\{E_k e^{-j\hat{\theta}_k} \mathbf{c}_k^{*'} \tilde{\mathbf{a}}_k^*\}] = \sigma_a^2 \|\mathbf{c}_k\|^2 \sin(\theta_k - \hat{\theta}_k). \quad (19.56)$$

□

Thus, the SG PLL algorithm is first order with a sinusoidal phase detector.

## 19.6. FURTHER READING

Several tutorial papers are available on general echo cancellation topics [13,14,15]. The digital subscriber loop echo canceler application is summarized in [3], with more details given in [10,4,16]. For the voiceband data modem application, the early papers by Weinstein are recommended [7] as well as the more recent article by Werner which proposes the passband transversal filter approach [8].

There are numerous techniques for speeding up adaptation of the echo canceler using more sophisticated adaptation algorithm. For some references, see Section 11.8. In data echo cancellation in particular, a significant factor slowing adaptation is the far-end data signal. This suggests another means of speeding adaptation, in which the data signal is adaptively removed from the cancellation error in a decision-directed fashion [17] in an approach called an *adaptive reference* canceler.

Another possibility is to use the least-square algorithm mentioned in Section 11.8. The data symbols can be chosen during a training period to assist in the canceler adaptation. In this case, the transmitted signal algebraic properties become much more important than the stochastic properties which we have emphasized in our convergence results. It has been shown that the mean values of the filter coefficients of a canceler based on least-squares can converge in  $N$  data symbols for an  $N$ -tap canceler [18]. Furthermore, it has been shown that the least-squares algorithm can be virtually as simple as the stochastic gradient algorithm for a reference signal which is chosen to be a pseudo-random sequence [19]. This sequence is also particularly simple to generate during a training period.

The implementation of a data echo canceler in monolithic form represents special challenges because of the high accuracy required. This is discussed in more depth in [10,4].

Some older work in speech cancelers and some more recent work in data cancelers has extended the adaptive echo canceler technique to nonlinear echo generation phenomena [20,16]. In data transmission, the objectives for degree of cancellation are sufficiently ambitious that nonlinear echo generation phenomena are of importance [10,21,4].

## APPENDIX 19-A REAL-ERROR CANCELER CONVERGENCE

In this appendix we analyze the convergence of the real error canceler for a passband channel and passband transversal filter. In general we will find that we must make stronger assumptions for this case to get simple results than we made in the complex error case. Specifically, we often have to assume independence of random variables whereas in the complex error case uncorrelated random variables will suffice.

### Minimum MSE Solution

We can find the minimum MSE solution for the real error canceler most easily by using the gradient formula derived in Exercise 19-3,

$$\nabla_c \left[ \text{Re} \{ E_k \} \right] = -2 \text{Re} \{ E_k \} \tilde{\mathbf{a}}_k^* . \quad (19.57)$$

Equating the expected value with the zero vector will give us the optimum coefficient vector. The evaluation of this expected value is aided by the following result:

### Exercise 19-7.

Given a sequence of transmitted rotated data symbol which are mutually independent, and for which the real and imaginary parts are zero-mean, identically distributed, and independent, show that

$$E [\tilde{\mathbf{a}}_k \tilde{\mathbf{a}}_k'] = \mathbf{0} . \quad (19.58)$$

Similarly, for the same assumptions on the uncancelable error  $V_k$ , show that

$$E [V_k^2] = 0 . \quad (19.59)$$

Lest this latter result seem strange, remember that  $V_k$  is complex valued, and hence its variance is  $E [|V_k|^2]$ , not  $E [V_k^2]$ .  $\square$

Using this result and taking the expected value of (19.57), we get immediately

$$\Phi(\tilde{\mathbf{h}} - \mathbf{c}) + \mathbf{p} + \mathbf{q} = \mathbf{0} \quad (19.60)$$

where  $\Phi$  and  $\mathbf{p}$  are defined as before and

$$\mathbf{q} = E [V_k^* \tilde{\mathbf{a}}_k^*] . \quad (19.61)$$

It follows that the optimum coefficient vector is

$$\mathbf{c}_{\text{opt}} = \tilde{\mathbf{h}} + \Phi^{-1}(\mathbf{p} + \mathbf{q}) \quad (19.62)$$

which is almost the same as for the complex error case with the addition of the  $\mathbf{q}$  term. In fact, for the important case where the uncancelable error is independent of the transmitted data symbols and both are zero-mean, (19.62) reduces to  $\mathbf{c}_{\text{opt}} = \tilde{\mathbf{h}}$  and the solution is the same as for the complex error case.

### Convergence of the SG Algorithm

The real error SG algorithm is given by (19.36). We can easily develop a stochastic difference equation governing the trajectory of the coefficient vector error.

#### Exercise 19-8.

In analogy to Exercise 19-4, show that for the real error algorithm we get a slightly more complicated result

$$\mathbf{q}_k = \Gamma_k \mathbf{q}_{k-1} - \Lambda_k \mathbf{q}_{k-1}^* + \beta \operatorname{Re}\{D_k\} \tilde{\mathbf{a}}_k^* \quad (19.63)$$

where the stochastic matrices are

$$\Gamma_k = \mathbf{I} - \frac{1}{2}\beta \tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k', \quad \Lambda_k = \frac{1}{2}\beta \tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k^{*'} \quad (19.64)$$

□

Fortunately, under reasonable assumptions the three terms in (19.63) are independent. First, from the orthogonality principle that the real error for the minimum MSE canceler  $\operatorname{Re}\{D_k\}$  is uncorrelated with the sequence of transmitted data symbols,

$$E[\operatorname{Re}\{D_k\} \tilde{\mathbf{a}}_k^*] = \mathbf{0}. \quad (19.65)$$

If we further assume that  $D_k$  is independent of the transmitted data symbols, and zero-mean, then the expected value of any cross terms between  $\operatorname{Re}\{D_k\}$  and  $\Gamma_k$  or  $\Lambda_k$  will be zero. Further,  $\Gamma_k$  and  $\Lambda_k$  are themselves uncorrelated.

#### Exercise 19-9.

Using the results of Exercise 19-7, show that

$$E[\Gamma_k \Lambda_k^*] = \mathbf{0}. \quad (19.66)$$

□

Now we are prepared to determine the expected value of  $\|\mathbf{q}_k\|^2$ , using in part the results of appendix 9-A. From that appendix (recall that the definition of  $\Gamma_k$  is slightly different here),

$$E[\Gamma_k^{*'} \Gamma_k] = (1 - \beta \sigma_a^2 + \frac{1}{4}\beta^2(\eta_a + (N-1)\sigma_a^4))\mathbf{I}, \quad \eta_a = E[|a_k|^4]. \quad (19.67)$$

By a similar computation, we can find the second term.

#### Exercise 19-10.

Show that approximately

$$E[\Lambda_k^{*'} \Lambda_k] = 0.25\beta^2(\eta_a + (N-1)\sigma_a^4)\mathbf{I}. \quad (19.68)$$

□

Finally, the expected norm-squared of (19.63) becomes the expected norm-squared of three terms,

$$\begin{aligned}
 E[\|\Gamma_k \mathbf{q}_{k-1}\|^2] &= \mathbf{q}_{k-1}^T E[\Gamma_k^{*'} \Gamma_k] \mathbf{q}_{k-1} \\
 &= (1 - \beta \sigma_a^2 + \frac{1}{4} \beta^2 (\eta_a + (N-1) \sigma_a^4)) \|\mathbf{q}_{k-1}\|^2
 \end{aligned} \quad (19.69)$$

$$E[\|\Lambda_k \mathbf{q}_{k-1}^*\|^2] = \mathbf{q}_{k-1}^T E[\Lambda_k^{*'} \Lambda_k] \mathbf{q}_{k-1} = \frac{1}{4} \beta^2 (\eta_a + (N-1) \sigma_a^4) \|\mathbf{q}_{k-1}\|^2 \quad (19.70)$$

$$E[\|\beta \operatorname{Re}\{D_k\} \hat{\mathbf{a}}_k^*\|^2] = \beta^2 E[(\operatorname{Re}\{D_k\})^2] \|\hat{\mathbf{a}}_k\|^2 = N \sigma_a^2 \beta^2 E[(\operatorname{Re}\{D_k\})^2] \quad (19.71)$$

Evaluation of this expression is aided by the following result.

#### Exercise 19-11.

- (a) Assume that the uncanceled error  $D_k$  consists of a filtered far-end data signal plus an additive noise. Further make the usual independence and white-noise assumptions on these two components and show that

$$E[D_k^2] = E[(D_k^*)^2] = 0. \quad (19.72)$$

- (b) Show that

$$E[(\operatorname{Re}\{D_k\})^2] = \frac{1}{2} E[|D_k|^2] = \frac{1}{2} \xi_{\min}. \quad (19.73)$$

In other words, the real and imaginary parts of  $E[|D_k|^2]$  are equal.  $\square$

These results give the following difference equation for the norm-squared error vector,

$$E[\|\mathbf{q}_k\|^2] = \gamma E[\|\mathbf{q}_{k-1}\|^2] + \frac{1}{2} N \sigma_a^2 \beta^2 \xi_{\min} \quad (19.74)$$

$$\gamma = 1 - \beta \sigma_a^2 + \frac{1}{4} \beta^2 (\eta_a + (N-1) \sigma_a^4) \quad (19.75)$$

Now we are in a position to compare the real error and complex error cancelers. First looking at the asymptotic MSE, the complex error case is given by (19.49), whereas from (19.74)

$$E[\|\mathbf{q}_k\|^2] \rightarrow \frac{N \beta}{2 - N \beta \sigma_a^2} \xi_{\min} \quad (19.76)$$

which is the same as (19.49). Similarly, calculating an approximate time constant  $\tau$  from (19.75) as  $\gamma^T = 1/e$ , we get for small step-size

$$\tau \approx \frac{1}{\beta \sigma_a^2} \quad (19.77)$$

which is twice as long as for the complex error canceler (19.46).



## PROBLEMS

- 19-1.** Consider using echo cancellation for a digital subscriber loop with AMI line coding (Section 12.1). What options are there for realization of the line coder, and where would it be most reasonable to connect the echo canceler input?
- 19-2.** Consider a V.32 voiceband data modem with the following parameters: baud rate 2400 Hz, carrier frequency 1800 Hz, passband channel covering the band from 300 to 3000 Hz. Further assume that the echo response has a duration of 32 baud intervals.
- Assuming the receive signal is sampled at a rate equal to an integer multiple of the baud rate, what is the most reasonable sampling rate? Discuss the considerations in the choice of this rate.
  - For this sampling rate, compare the multiplication rate for two cancelers, one connected to the sampled transmitted waveform and the other to the transmitted data symbols.
- 19-3.** For a passband echo canceler, it is possible to put a demodulator in the receiver prior to cancellation of the echo.
- Show two alternative configurations, one using a phase splitter and the other a lowpass filter. Develop an equivalent echo channel model analogous to Figure 19-7.
  - Is it possible or reasonable to consider a real-error canceler for this configuration?
  - Describe the echo canceler required for this configuration.
  - How will the adaptation rate of this configuration compare to the configurations considered in Section 19.3?
  - Discuss the advantages and disadvantages of this configuration.
- 19-4.** Assuming the echo response extends for  $N$  baud intervals and  $R$  interleaved cancelers, compare the complexity as measured in equivalent real-valued multiplication rates for all combinations of a baseband and passband transversal filter with a real-error and complex-error canceler. Which configurations are more attractive in accordance with this complexity metric?
- 19-5.** How would you modify Figure 19-10 to use a complex-error canceler? Show that only one phase splitter is required, at the expense of a second lowpass filter.
- 19-6.** Determine the MSE solution for the complex-error canceler with baseband transversal filter, and find the optimal coefficient vector. **HINT:** Show that minimizing  $E[|E_k|^2]$  is equivalent to minimizing  $E[|e^{-j\omega_k(k + \frac{1}{R})T} E_k|^2]$ , and then minimize the latter quantity.
- 19-7.** For a passband echo channel, we can use a baseband echo canceler followed by modulator, or a modulator (rotator) followed by passband echo canceler. Give a convincing argument that the convergence rate and asymptotic MSE of the baseband echo canceler is the same as the convergence rate and MSE of the passband echo canceler.
- 19-8.**
- The PLL algorithm of Exercise 19-6 will have a tracking capability somewhat dependent on the echo impulse response. Explain.
  - How would you fix this problem?
- 19-9.**
- Show that

$$\frac{\partial(\text{Re}\{E_k\})^2}{\partial\theta} = -2\text{Re}\{E_k\} \text{Im}\{e^{-j\theta} \mathbf{c}_k^* \mathbf{a}_k^*\}. \quad (19.78)$$

- Use this result to develop a first-order PLL algorithm that uses only the real-error. Interpret this algorithm graphically.

- (c) Find the expected value of the phase correction term.

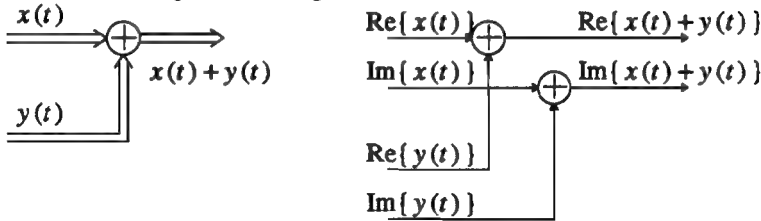
## REFERENCES

1. B. Aschrafi, P. Meschkat, and K. Szechenyi, "Field Trial of a Comparison of Time Separation, Echo Cancellation, and Four-Wire Digital Subscriber Loops," *Proceedings of the Int. Symp. on Subscriber Loops and Services*, (Sep. 1982).
2. J-O. Andersson, B. Carlquist, A. Bauer, and I. Dahlqvist, "An LSI Implementation of an ISDN Echo Canceller: Design and Network Aspects," *IEEE Journal on Selected Areas in Communications*, (this issue).
3. D. G. Messerschmitt, "Design Issues for the ISDN U-Interface Transceiver," *IEEE Jour. on Special Areas in Communications*, (Nov. 1986).
4. O. Agazzi, D. A. Hodges, and D. G. Messerschmitt, "Large-Scale Integration of Hybrid-Method Digital Subscriber Loops," *IEEE Trans. on Communications* COM-30 p. 2095 (Sep. 1982).
5. J. Tzeng, D. Hodges, and D. G. Messerschmitt, "Baud Rate Timing Recovery in Digital Subscriber Loops," *IEEE Int. Conf. on Communications*, (June 1985).
6. K. H. Mueller and M. Muller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Trans. on Communications* COM-24 pp. 516-531 (May 1976).
7. S. B. Weinstein, "A Passband Data-Driven Echo Canceller for Full-Duplex Transmission on Two-Wire Circuits," *IEEE Trans. on Communications*, (July 1977).
8. J. J. Werner, "An Echo-Cancellation-Based 4800 Bit/s Full-Duplex DDD Modem," *IEEE Journal on Selected Areas in Communications* SAC-2(5)(Sep. 1984).
9. B. Widrow, J. McCool, M. Larimore, and C. Johnson, Jr., "Stationary and Non-Stationary Learning Characteristics of the LMS Adaptive Filter," *Proc. IEEE* 64(8) pp. 1151-1162 (Aug. 1976).
10. N. A. M. Verhoeckx, H. C. Van Den Elzen, F. A. M. Sniijders, and P. J. Van Gerwen, "Digital Echo Cancellation for Baseband Data Transmission," *IEEE Trans. on ASSP* ASSP-27(6)(Dec. 1979).
11. D. L. Duttweiler, "A Twelve-Channel Digital Echo Canceller," *IEEE Trans. on Communications*, pp. 647-653 (May 1978).
12. M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*, Kluwer Academic Publishers, Boston (1984).
13. M. Sondhi and D. A. Berkley, "Silencing Echos on the Telephone Network," *IEEE Proceedings* 8(Aug. 1980).
14. D. G. Messerschmitt, "Echo Cancellation in Speech and Data Transmission," *IEEE Jour. on Selected Areas in Communications* SAC-2(2) p. 283 (March 1984).
15. D. G. Messerschmitt, "Echo Cancellation in Speech and Data Transmission," pp. 182 in *Advanced Digital Communications Systems and Signal Processing Techniques*, ed. K. Feher, Prentice-Hall, Englewood Cliffs, N.J. (1987).
16. O. Agazzi, D. G. Messerschmitt, and D. A. Hodges, "Nonlinear Echo Cancellation of Data Signals," *IEEE Trans. on Communications* COM-30 p. 2421 (Nov. 1982).
17. D. D. Falconer, "Adaptive Reference Echo Cancellation," *IEEE Trans. on Communications* COM-30(9)(Sept. 1982).
18. T. L. Lim and M. S. Mueller, "Rapid Equalizer Start-Up Using Least Squares Algorithms," 1980 *Proc. IEEE ICC*, ().

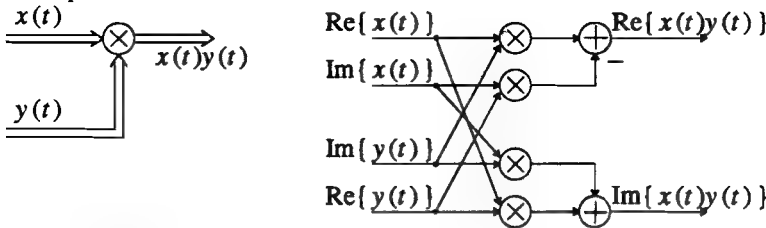
19. J. Salz, "On the Start-Up Problem in Digital Echo Cancellers," *BSTJ* **60**(10) pp. 2345-2358 (July-Aug., 1983).
20. E. J. Thomas, "Some considerations on the application of the Volterra representation of non-linear networks to adaptive echo cancellers," *BSTJ* **50**(8) pp. 2797-2805 (Oct. 1971).
21. N. Holte and S. Stueflotten, "A New Digital Echo Canceler for Two-Wire Subscriber Lines," *IEEE Trans. on Communications* **COM-29**(11) pp. 1573-1581 (Nov. 1981).

## EXERCISE SOLUTIONS

- 2-1. Addition of two complex-valued signals is illustrated below:

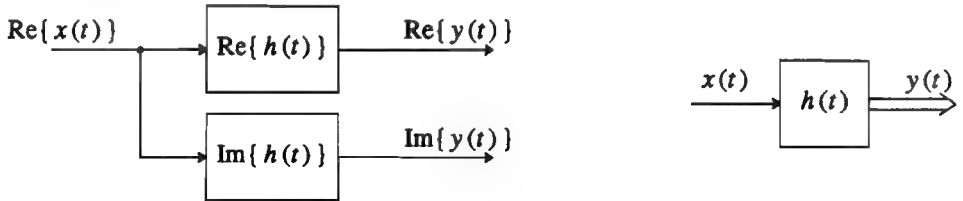


and multiplication below:

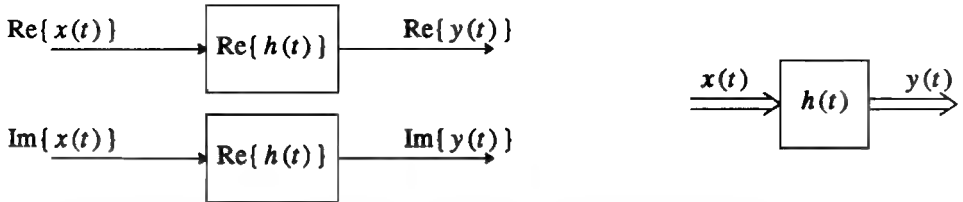


Complex addition is accomplished by two real additions, and complex multiplication by four real multiplications and two real additions.

- 2-2. A complex system with a real-valued input:



A real system with a complex-valued input:



- 2-3. We can treat the convolution  $x(t) * h(t)$  just like complex multiplication, since the convolution operation is linear — an integration. To check linearity, for a complex constant  $A$  and two input signals  $x_1(t)$  and  $x_2(t)$ ,

$$(x_1(t) + A \cdot x_2(t)) * h(t) = x_1(t) * h(t) + A \cdot (x_2(t) * h(t)) \quad (2.112)$$

following the rules of complex arithmetic. This establishes linearity.

2-4.

$$Y(j\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau e^{-j\omega t} dt. \quad (2.113)$$

Observe that

$$e^{-j\omega t} = e^{-j\omega\tau} e^{-j\omega(t-\tau)} \quad (2.114)$$

so that

$$\begin{aligned} Y(j\omega) &= \int_{-\infty}^{\infty} h(\tau) e^{-j\omega\tau} d\tau \int_{-\infty}^{\infty} x(t-\tau) e^{-j\omega(t-\tau)} dt \\ &= H(j\omega)X(j\omega) \end{aligned} \quad (2.115)$$

after a change of variables.

2-5. Take the Fourier transform of both sides of (2.2), getting

$$\begin{aligned} \hat{X}(j\omega) &= \int_{-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x_m \delta(t-mT) e^{-j\omega t} dt \\ &= \sum_{m=-\infty}^{\infty} x_m \int_{-\infty}^{\infty} \delta(t-mT) e^{-j\omega t} dt \\ &= \sum_{m=-\infty}^{\infty} x_m e^{-j\omega mT} = X(e^{j\omega T}). \end{aligned} \quad (2.116)$$

2-6. The impulse response of the system is

$$f_k = f(kT) \quad (2.117)$$

and hence (2.17) gives the frequency response directly,

$$F(e^{j\omega T}) = \frac{1}{T} \sum_m F[j(\omega + m\frac{2\pi}{T})]. \quad (2.118)$$

2-7. Given  $X(j\omega) = 0$  for all  $|\omega| > \pi/T$ , (2.17) implies that

$$X(e^{j\omega T}) = \frac{1}{T} X(j\omega) \quad \text{for all } |\omega| < \frac{\pi}{T}. \quad (2.119)$$

To get  $x(t)$  from  $x_k$ , therefore, we can use

$$F(j\omega) = \begin{cases} T; & |\omega| < \pi/T \\ 0; & \text{otherwise} \end{cases} \quad (2.120)$$

in Figure 2-1.

2-8. The transfer function of the filter proposed as a phase splitter is

$$\frac{1}{2}(1 + j \cdot H(j\omega)) = \frac{1}{2}(1 + j(-j \operatorname{sgn}(\omega))) = \frac{1}{2}(1 + \operatorname{sgn}(\omega)). \quad (2.121)$$

As expected this transfer function is unity for positive frequencies and zero for negative frequencies.

2-9. The modulus-squared of complex-valued signal  $u(t)$  is

$$|u(t)|^2 = \frac{1}{2}\{y^2(t) + \dot{y}^2(t)\} \quad (2.122)$$

and since the Hilbert transform is a phase-only filter, the energy of  $y(t)$  and its Hilbert transform  $\dot{y}(t)$  are the same.

2-10. First show that  $S < \infty$  implies BIBO. Suppose the input is bounded by  $x_k \leq L$ . Then

$$|y_k| = \left| \sum_{m=-\infty}^{\infty} h_m x_{k-m} \right| \leq L \sum_{m=-\infty}^{\infty} |h_m| = LS < \infty. \quad (2.123)$$

Then show that if  $S = \infty$  there exists a bounded input such that the output is unbounded. Such an input is

$$x_k = \begin{cases} h_{-k}^* / |h_{-k}|; & k \text{ such that } h_k \neq 0 \\ 0; & k \text{ such that } h_k = 0 \end{cases} \quad (2.124)$$

2-11.

(a) If the sequence is left-sided (2.30) becomes

$$\sum_{k=-\infty}^K |h_k| |z|^{-k} < \infty. \quad (2.125)$$

for some  $K$ . This sum can be rewritten

$$|z|^{-K} \sum_{k=0}^{\infty} |h_{K-k}| \cdot |z|^k \quad (2.126)$$

and we recognize that the  $|z|^{-K}$  cannot affect convergence except at  $|z| = 0$  and  $|z| = \infty$ . All the terms in the sum are positive powers of  $|z|$ , and hence if they converge for some  $|z| = R$  they must converge for all smaller  $|z|$  (except possibly  $|z| = 0$ ).

(b) If  $K > 0$ , the sequence is not anti-causal, and there is a  $K$ -th order pole at  $z = 0$ , the ROC does not include  $z = 0$ . If  $K = 0$ , then  $H(0) = h_K$ , the ROC includes  $z = 0$ . If  $K < 0$ , there is a  $K$ -th order zero at  $z = 0$ , which is therefore included in the ROC.

2-12. This follows directly from the observation that  $x_{k-l}$  for a fixed integer  $l$  has Z transform  $z^{-l}X(z)$ . Taking the Z transform of both sides of (2.34), we get the desired results.

2-13. The zero vector is

$$\mathbf{0} \leftrightarrow (\cdots 0, \cdots, 0, \cdots) \quad (2.127)$$

or

$$\mathbf{0} \leftrightarrow y(t) = 0. \quad (2.128)$$

The rest is a tedious but straightforward verification of the properties.

2-14. This is a straightforward evaluation. For example,

$$\langle Y, X \rangle = \int_{-\infty}^{\infty} y(t) x^*(t) dt = \left( \int_{-\infty}^{\infty} x(t) y^*(t) dt \right)^* = \langle X, Y \rangle^*. \quad (2.129)$$

2-15. Let  $Y \in M$ , then

$$\begin{aligned} \|X - Y\|^2 &= \|X - P_M(X) + P_M(X) - Y\|^2 \\ &= \|X - P_M(X)\|^2 + \|P_M(X) - Y\|^2 + 2\langle X - P_M(X), P_M(X) - Y \rangle \end{aligned} \quad (2.130)$$

Since  $(P_M(X) - Y) \in M$  and  $(X - P_M(X))$  is orthogonal to the subspace  $M$ , the last term is 0 and

$$\begin{aligned}\|X - Y\|^2 &= \|X - P_M(X)\|^2 + \|P_M(X) - Y\|^2 \\ &\geq \|X - P_M(X)\|^2\end{aligned}\quad (2.131)$$

with equality if and only if  $Y = P_M(X)$ .

- 2-16. The inequality is obviously true (with equality) if  $X = 0$  or  $Y = 0$ , so assume that  $X \neq 0$  and  $Y \neq 0$ . Then we have the inequality

$$\begin{aligned}0 &\leq \|X - \alpha Y\|^2 \\ 0 &\leq \|X\|^2 - 2\operatorname{Re}\{\alpha^* \langle X, Y \rangle\} + |\alpha|^2 \|Y\|^2\end{aligned}\quad (2.132)$$

If we let

$$\alpha = \frac{\langle X, Y \rangle}{\|Y\|^2} \quad (2.135)$$

then the previous inequality becomes

$$0 \leq \|X\|^2 - \frac{|\langle X, Y \rangle|^2}{\|Y\|^2} \quad (2.136)$$

from which the Schwarz inequality follows immediately.

2-17.

- (a) If  $S(e^{j\omega T})$  is real valued for all  $\omega$ ,  $S(e^{j\omega T}) = S^*(e^{j\omega T})$ . It is easy to show that the inverse DTFT of  $S^*(e^{j\omega T})$  is  $s_{-k}^*$ , from which the result follows.
- (b) It is easy to show that the Z transform of  $s_{-k}^*$  is  $S^*(1/z^*)$ , so taking Z transforms on both sides of  $s_k = s_{-k}^*$ , the result follows.

- 2-18. Noting that, since  $H_{\text{zero}}(z)$  is monic and causal,

$$B \cdot z^L H_{\text{zero}}(z) = B \cdot z^L \prod_{i=1}^K (1 - e^{j\theta_i} z^{-1}) \quad (2.137)$$

and by exercise 2-17b if this term is to be real valued on the unit circle is must equal its own conjugate evaluated at  $1/z^*$ . Noting that

$$B^* z^{-L} H_{\text{zero}}(1/z^*) = B^* z^{K-L} \exp\{-j \sum_{i=1}^K \theta_i\} (-1)^K \prod_{i=1}^K (1 - e^{j\theta_i} z^{-1}) \quad (2.138)$$

Equating (2.137) and (2.138), we conclude several facts. First, equating powers of  $z$  we conclude that we must have  $K = 2L$ , one consequence of which is an even number of zeros. Then  $(-1)^K = 1$ , and equating the constant multipliers,

$$\frac{B}{B^*} = \exp\{-j \sum_{i=1}^{2L} \theta_i\} \quad (2.139)$$

which implies that  $B$  can have any magnitude, but must have a particular phase given by (2.90).

- 2-19. Evaluating the transfer function on the unit circle,

$$\begin{aligned}B \cdot z^L H_{\text{zero}}(z) \Big|_{z=e^{j\omega}} &= C \exp\{-j \sum_{i=1}^{2L} \theta_i/2\} e^{j\omega L} \prod_{i=1}^{2L} (1 - e^{-j(\omega - \theta_i)}) \\ &= C \cdot 2^{2L} (-1)^L \prod_{i=1}^{2L} \sin\left(\frac{\omega - \theta_i}{2}\right).\end{aligned}\quad (2.140)$$

When all zeros are double, assuming  $\theta_i = \theta_{i+L}$ ,  $1 \leq i \leq L$ , this reduces to

$$B e^{j\omega L} H_{\text{zero}}(e^{j\omega}) = C \cdot 2^{2L} (-1)^L \prod_{i=1}^L \sin^2\left(\frac{\omega - \theta_i}{2}\right) \quad (2.141)$$

which will be non-negative if  $C$  is chosen with the proper sign. Conversely, if there is a simple zero at  $z = e^{j\theta_i}$ , then choose a constant  $\epsilon$  small enough that the interval  $[\theta_i - \epsilon, \theta_i + \epsilon]$  does not overlap any of the other  $\theta_i$ . Then the term  $\sin(\frac{\omega - \theta_i}{2})$  changes sign in this interval, while all the other terms have the same sign in this interval. It follows that  $H_{\text{zero}}(e^{j\omega})$  must change sign in this interval, and hence cannot be consistently non-negative.

3-1. This follows from

$$\begin{aligned} E[e^{s(X+Y)}] &= E[e^{sX}]E[e^{sY}] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{sx} f_X(x) e^{sy} f_Y(y) dx dy \\ &= \int_{-\infty}^{\infty} e^{sx} f_X(x) dx \int_{-\infty}^{\infty} e^{sy} f_Y(y) dy = \Phi_X(s) \Phi_Y(s) . \end{aligned} \quad (3.201)$$

3-2. Evaluating the derivatives,

$$\Phi_X(s) \big|_{s=0} = 1 , \quad (3.202)$$

$$\frac{\partial}{\partial s} \Phi_X(s) \big|_{s=0} = E[X e^{sX}] \big|_{s=0} = E[X] , \quad (3.203)$$

$$\frac{\partial^2}{\partial s^2} \Phi_X(s) \big|_{s=0} = E[X^2 e^{sX}] \big|_{s=0} = E[X^2] . \quad (3.204)$$

3-3.

(a) The distribution function can be written in terms of a unit step function  $u(x)$  as

$$1 - F_X(x) = \int_{-\infty}^{\infty} u(y-x) f_X(y) dy = \int_x^{\infty} f_X(y) dy \quad (3.205)$$

and since  $u(y-x)$  is bounded by  $e^{(y-x)s}$  for  $s \geq 0$ ,

$$1 - F_X(x) \leq \int_{-\infty}^{\infty} e^{(y-x)s} f_X(y) dy = e^{-sx} \Phi_X(s) . \quad (3.206)$$

(b) Obtained by a similar technique.

(c) Take the derivative of the bound with respect to  $s$  and set to zero.

3-4. Suppose that  $y$  is a discrete value that  $Y$  takes on with probability  $a$ . Then

$$f_Y(\alpha) = a \delta(\alpha - y) . \quad (3.207)$$

Integrate (3.30) over small intervals about  $y$ , or over  $(y - \epsilon, y + \epsilon)$  for small enough  $\epsilon$ . Equation (3.32) follows similarly, or it can be easily derived from the definition of conditional probabilities (3.27).

3-5. By direct calculation we have

$$\Pr[X > x] = \frac{1}{\sigma\sqrt{2\pi}} \int_x^{\infty} e^{-(\alpha - \mu)^2/2\sigma^2} d\alpha = \frac{1}{\sigma\sqrt{2\pi}} \int_{(x-\mu)/\sigma}^{\infty} e^{-w^2/2} \sigma dw = Q\left[\frac{x - \mu}{\sigma}\right] , \quad (3.208)$$

where we have used the change of variables  $w = (\alpha - \mu)/\sigma$ .



3-6.

- (a) The moment generating function can be obtained by evaluating the integral

$$\Phi_X(s) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{sx} e^{-(x-\mu)^2/2\sigma^2} dx. \quad (3.209)$$

Combining the two exponents and completing the square in the resulting exponent in the integral, it becomes  $-\left((x-a)^2 - b\right)/2\sigma^2$  where  $b = 2\mu\sigma^2 s + \sigma^4 s^2$ . The  $e^{b/2\sigma^2}$  term can be taken outside the integral, and the remaining integrand is just a Gaussian density function and therefore integrates to unity. Thus, the moment generating function is  $\Phi_X(s) = e^{b/2\sigma^2}$ , and substituting for  $b$  we get the claimed result.

- (b) The value of
- $s$
- giving the tightest bound can be solved as

$$s = \frac{x - \mu}{\sigma^2} \quad (3.210)$$

and hence the bound is valid as long as  $x \geq \mu$ . Substituting this value of  $s$  into the bound, we get

$$1 - F_X(x) \leq e^{-(x-\mu)^2/2\sigma^2} \quad (3.211)$$

which looks remarkably like the Gaussian density. Note that when  $x = \mu$ , the actual probability is  $1/2$  and the Chernoff bound is  $e = 2.28$ , so the bound is rather loose. It becomes much tighter for larger values of  $x$ . The relation (3.43) follows by letting  $\mu = 0$  and  $\sigma = 1$ .

- 3-7. Consider a scaled Gaussian,
- $Y = aX$
- . If the variance of
- $X$
- is
- $\sigma^2$
- , then the variance of
- $Y$
- is
- $a^2\sigma^2$
- . Hence the moment generating function of
- $Y$
- is

$$\Phi_Y(s) = e^{a^2\sigma^2 s^2/2}. \quad (3.212)$$

The moment generating function is

$$\Phi_Z(s) = e^{(a_1^2 + \dots + a_n^2)\sigma^2 s^2/2}. \quad (3.213)$$

This is the moment generating function of a zero mean Gaussian random variable with variance (3.46).

- 3-8. We only need to show

$$E[XY] = E[X]E[Y] \Rightarrow f_{X,Y}(x,y) = f_X(x)f_Y(y). \quad (3.214)$$

From (3.48) and the fact that the random variables have zero-mean,  $\mu = 0$ . Now (3.47) is easily factored into two parts.

- 3-9. Define

$$X + Y \leftrightarrow X + Y \quad (3.215)$$

$$\alpha X \leftrightarrow \alpha X \quad (3.216)$$

$$0 \leftrightarrow 0 \text{ (the zero random variable)} \quad (3.217)$$

$$-X \leftrightarrow -X \quad (3.218)$$

With these definitions, verification of the properties is straightforward, relying on similar linearity properties of random variables.

3-10. This is very straightforward. For example,

$$\langle Y, X \rangle = E[YX^*] = \{E[XY^*]\}^* = \langle X, Y \rangle^* \quad (3.219)$$

3-11. First show that

$$R_W(\tau) = E[W(t + \tau)W^*(t)] = h^*(-\tau) * R_{WX}(\tau) \quad (3.220)$$

by substituting for one of the  $W(t)$  in terms of  $X(t)$ . Then show that

$$R_{WX}(\tau) = h(\tau) * R_X(\tau), \quad (3.221)$$

completing the first result. Finally, show that the Fourier transform of  $h^*(-\tau)$  is  $H^*(j\omega)$ , and that the Laplace transform of  $h^*(-\tau)$  is  $H^*(-s^*)$ . Both of these are easily done using the Fourier and Laplace transform definitions.

3-12. Calculating first the cross-correlation of the input and output,

$$R_{XW}(m) = E[X_{k+m}W_k^*] = E[X_{k+m} \sum_{n=-\infty}^{\infty} X_n^* h_{k-n}^*] = R_X(m) * h_{-m}^*. \quad (3.222)$$

Then calculating the output correlation function,

$$R_W(m) = E[W_{k+m}W_k^*] = E[\sum_{n=-\infty}^{\infty} X_n h_{k+m-n} W_k^*] = R_{XW}(m) * h_m. \quad (3.223)$$

Finally, show that the Fourier transform of  $h_{-m}^*$  is  $H^*(e^{j\omega T})$  and the Z transform is  $H^*(1/z^*)$ .

3-13. First we can calculate that

$$R_{WY}(\tau) = R_{XY}(\tau) * h(\tau) \quad (3.224)$$

and then

$$R_{WU}(\tau) = R_{WY}(\tau) * g^*(-\tau) = R_{XY}(\tau) * h(\tau) * g^*(-\tau). \quad (3.225)$$

Taking the Fourier transform we get the desired result.

3-14. Write

$$\hat{H}(z) = KH_{\text{allpass}}(z), \quad (3.226)$$

where  $H_{\text{allpass}}(z)$  is an allpass filter. Modifying (2.41) slightly, write

$$H(z) = \frac{Kz^{-N}A(z)}{z^{-M}A^*(1/z^*)}, \quad A(z) = 1 + a_1z + \cdots + a_Nz^N, \quad (3.227)$$

and note that the denominator is causal if and only if  $M = 0$ , in which case it will also be monic. Thus  $H(z)$  will be causal and monic if and only if the numerator is causal and monic. The numerator is causal, but it will be monic if and only if  $Ka_N = 1$ , or

$$K = \frac{1}{a_N}. \quad (3.228)$$

From (2.33) it is easy to see that

$$|a_N| = \prod_{k=1}^N |d_k|, \quad (3.229)$$

where  $\{d_k; k \leq N\}$  is the set of poles of the filter. Since the filter is causal, it will be stable if and only if all its poles have magnitude less than one. This implies that  $|a_N| < 1$  or  $K > 1$ . Thus the gain of the filter must be greater than one.

- 3-15. This relation can be obtained by exactly the same method as Appendix 3-A, although it is tedious.
- 3-16. We use the fact that the next state  $\Psi_{k+1}$  of a Markov chain is independent of the past states  $\Psi_{k-1}, \Psi_{k-2}, \dots$  given the present state  $\Psi_k$  to show that *all* future samples of the Markov chain are independent of the past given knowledge of the present.

We wish to show that for any  $n > 0$  and any  $k$ ,

$$p(\Psi_{k+n} | \Psi_k, \Psi_{k-1}, \dots) = p(\Psi_{k+n} | \Psi_k). \quad (3.230)$$

This is easily shown by induction. Observe that it is true for  $n = 1$ , by the definition of Markov chains (3.91). We can assume that it is true for some  $n$  and show it is true for  $n+1$ . A fact about conditional probabilities similar to that in (3.33) tells us that

$$\begin{aligned} p(\Psi_{k+n+1} | \Psi_k, \Psi_{k-1}, \dots) \\ = \sum_{\Psi_{k+1} \in \Omega_T} p(\Psi_{k+n+1} | \Psi_{k+1}, \Psi_k, \Psi_{k-1}, \dots) p(\Psi_{k+1} | \Psi_k, \Psi_{k-1}, \dots). \end{aligned} \quad (3.231)$$

Since we assume that (3.230) is true for  $n$ ,

$$p(\Psi_{k+n+1} | \Psi_{k+1}, \Psi_k, \Psi_{k-1}, \dots) = p(\Psi_{k+n+1} | \Psi_{k+1}). \quad (3.232)$$

It is also therefore true that

$$p(\Psi_{k+n+1} | \Psi_{k+1}, \Psi_k, \Psi_{k-1}, \dots) = p(\Psi_{k+n+1} | \Psi_{k+1}, \Psi_k). \quad (3.233)$$

Furthermore, from the definition of Markov chains,

$$p(\Psi_{k+1} | \Psi_k, \Psi_{k-1}, \dots) = p(\Psi_{k+1} | \Psi_k). \quad (3.234)$$

Substituting (3.233) and (3.234) into (3.231) we get

$$p(\Psi_{k+n+1} | \Psi_k, \Psi_{k-1}, \dots) = \sum_{\Psi_{k+1} \in \Omega_T} p(\Psi_{k+n+1} | \Psi_{k+1}, \Psi_k) p(\Psi_{k+1} | \Psi_k). \quad (3.235)$$

Using the same fact about conditional probabilities (3.33) we can eliminate the summation to get

$$p(\Psi_{k+n+1} | \Psi_k, \Psi_{k-1}, \dots) = p(\Psi_{k+n+1} | \Psi_k), \quad (3.236)$$

which shows that (3.92) is valid for  $n+1$ .

- 3-17. Multiplying both sides of (3.96) by  $z^{-k}$  and summing from  $k = 0$  to  $k = \infty$ ,

$$\sum_{k=0}^{\infty} p_{k+1}(j) z^{-k} = \sum_{i \in \Omega_T} p(j|i) \sum_{k=0}^{\infty} p_k(i) z^{-k}.$$

Changing variables and letting  $m = k+1$ ,

$$\sum_{m=0}^{\infty} p_m(j) z^{-m+1} = \sum_{i \in \Omega_T} p(j|i) P_i(z)$$

or

$$z(P_j(z) - p_0(j)) = \sum_{i \in \Omega_T} p(j|i) P_i(z)$$

3-18. We have

$$\begin{aligned} f_N &= \sum_{k=-\infty}^{\infty} kq_k(N) \\ &= \sum_{k=-\infty}^{\infty} kq_k(N)z^{-k} \Big|_{z=1}. \end{aligned} \quad (3.237)$$

But the latter summation can be evaluated using a derivative,

$$\frac{\partial}{\partial z} Q_N(z) = \sum_{k=-\infty}^{\infty} (-k)q_k(N)z^{-k-1} = -z^{-1} \sum_{k=-\infty}^{\infty} kq_k(N)z^{-k}. \quad (3.238)$$

The result follows immediately.

3-19.

(a) Given the power series

$$e^a = \sum_{k=0}^{\infty} \frac{a^k}{k!} \quad (3.239)$$

and differentiating it once

$$e^a = \frac{1}{a} \sum_{k=1}^{\infty} \frac{ka^k}{k!} \quad (3.240)$$

and differentiating it twice

$$e^a = \frac{1}{a^2} \left[ \sum_{k=1}^{\infty} \frac{k^2 a^k}{k!} - \sum_{k=1}^{\infty} \frac{ka^k}{k!} \right]. \quad (3.241)$$

The moments follow immediately.

(b) The moment generating function is

$$\Phi_N(s) = e^{-a} \sum_{k=-\infty}^{\infty} \frac{(ae^s)^k}{k!} = e^{ae^s - a} \quad (3.242)$$

and taking the logarithm we get (3.121).

3-20. For these initial conditions, we get

$$q_k(t_0) = 1 \quad (3.243)$$

and the Laplace transform becomes

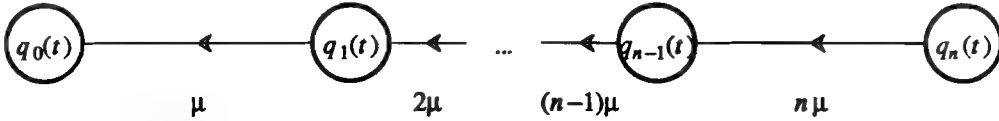
$$\begin{aligned} sQ_j(s) + \lambda Q_j(s) &= \lambda Q_{j-1}(s), \quad j \neq k \\ sQ_k(s) - q_k(t_0)e^{-st_0} + \lambda Q_k(s) &= \lambda Q_{k-1}(s). \end{aligned} \quad (3.244)$$

By iteration, we can establish that  $Q_j(s) = 0$  for  $j < k$  and for  $j \geq k$

$$Q_j(s) = \frac{\lambda^{j-k}}{(s + \lambda)^{j-k+1}} e^{-st_0}. \quad (3.245)$$

The result follows immediately by taking the inverse Laplace transform.

3-21. The state transition diagram is shown in the following figure:



The equations become for this case

$$\begin{aligned} \frac{dq_n(t)}{dt} + n\mu q_n(t) &= 0 \\ \frac{dq_j(t)}{dt} + j\mu q_j(t) &= (j+1)\mu q_{j+1}(t), \quad 0 \leq j < n \end{aligned} \quad (3.246)$$

with initial condition

$$q_j(0) = \begin{cases} 0, & 0 \leq j < n \\ 1, & j = n \end{cases} \quad (3.247)$$

Taking the Laplace transform, we get

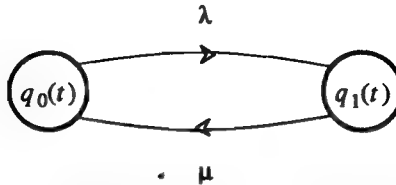
$$\begin{aligned} sQ_n(s) - q_n(0) + n\mu Q_n(s) &= 0 \\ sQ_j(s) + j\mu Q_j(s) &= (j+1)\mu Q_{j+1}(s), \quad 0 \leq j < n. \end{aligned} \quad (3.248)$$

It follows that

$$\begin{aligned} q_n(t) &= e^{-n\mu t} \\ q_j(t) &= (j+1)\mu e^{-j\mu t} * q_{j+1}(t) \end{aligned} \quad (3.249)$$

and the reader can verify by induction that (3.123) is valid.

3-22. This is a two-state birth and death process with transition diagram shown in the following figure:



State 0 corresponds to the server idle, and state 1 corresponds to the server busy. When the system is in state  $j = 0$ , arrivals occur at rate  $\lambda$ , and there are no departures because the server is idle. Similarly, in state  $j = 1$ , where the single server is busy, arrivals immediately depart from the system, have no effect, and therefore are not reflected on the state transition diagram. Also in this state, departures occur with rate  $\mu$  due to the completion of service. The differential equations governing the system are

$$\frac{dq_0(t)}{dt} = \mu q_1(t) - \lambda q_0(t) \quad (3.250)$$

$$\frac{dq_1(t)}{dt} = \lambda q_0(t) - \mu q_1(t). \quad (3.251)$$

Since the sum of the two probabilities must be unity,

$$q_0(t) + q_1(t) = 1, \quad (3.252)$$

we can clear one variable to yield a single differential equation for  $q_0(t)$ ,

$$\frac{dq_0(t)}{dt} + (\lambda + \mu)q_0(t) = \mu \quad (3.253)$$

and the result follows immediately by taking the Laplace transform.

3-23. Rewriting (3.128), we get

$$\mu q_{j+1} - \lambda q_j = \mu q_j - \lambda q_{j-1} \quad (3.254)$$

and since this recursion starts at zero,

$$q_{j+1} = \rho q_j = \rho^{j+1} q_0. \quad (3.255)$$

Using the fact that the probabilities must sum to unity, we can find that  $q_0 = 1 - \rho$ .

3-24. We get

$$\frac{\partial}{\partial s} \Phi_{X(t)}(s) = \Phi_{X(t)}(s) \lambda(t) * (h(t) e^{sh(t)}) \quad (3.256)$$

and setting  $s = 0$  we get the mean of (3.138). Taking the second derivative, we get

$$\frac{\partial^2}{\partial s^2} \Phi_{X(t)}(s) = \frac{\partial}{\partial s} \Phi_{X(t)}(s) \lambda(t) * (h(t) e^{sh(t)}) + \Phi_{X(t)}(s) \lambda(t) * (h^2(t) e^{sh(t)}) \quad (3.257)$$

and setting  $s = 0$  we get the second moment. Subtracting the square of the mean, we get the variance of (3.139).

3-25. The exact moment generating function is

$$\begin{aligned} \log_e \Phi_{X(t)}(s) &= \beta \lambda_0(t) * (e^{sh_0(t)\sqrt{\beta}} - 1) \\ &= \beta \lambda_0(t) * \left( \frac{sh_0(t)}{\sqrt{\beta}} + 0.5 \frac{s^2 h_0^2(t)}{\beta} \right) \end{aligned} \quad (3.258)$$

where all the other terms are of order  $1/\sqrt{\beta}$  or smaller and become insignificant as  $\beta \rightarrow \infty$ .

3-26. Note that

$$E[G_m G_n] = \begin{cases} E[G^2], & m = n \\ E[G]^2, & m \neq n \end{cases} \quad (3.259)$$

and also note that from Campbell's theorem

$$E\left[\sum_m h^2(t - t_m)\right] = \lambda(t) * h^2(t) = \sigma_X^2(t) \quad (3.260)$$

since this is a shot noise process with impulse response  $h^2(t)$ . The rest is straightforward but tedious algebra.

3-27. Part (a) is straightforward, so let's concentrate on part (b). Integrating,

$$e^{A(t)} x(t) - e^{A(t_0)} x(t_0) = \int_{t_0}^t b(u) e^{A(u)} du. \quad (3.261)$$

The solution of (3.174) follows directly from the observation that

$$\Lambda(t) = A(t) - A(t_0) \quad (3.262)$$

and some algebraic manipulation.

- 3-28. Noting that (3.134) is true for  $j = 0$ , assume it true for  $j$ . Then using (3.176) to determine the distribution for  $j+1$ ,

$$q_{j+1}(t) = e^{-\Lambda(t)} \int_{i_0}^t \lambda(u) \frac{\Lambda^j(u)}{j!} du. \quad (3.263)$$

But noting that  $\dot{\Lambda}(t) = \lambda(t)$ , this becomes

$$q_{j+1}(t) = e^{-\Lambda(t)} \int_{i_0}^t \dot{\Lambda}(u) \frac{\Lambda^j(u)}{j!} du \quad (3.264)$$

which can be integrated directly because it is a perfect differential to yield (3.134) for  $j+1$ .

- 4-1. Let the  $K$  outcomes have probabilities  $p_i$ ,  $1 \leq i \leq K$ . Using the inequality

$$\sum_i p_i \log_2 \frac{1}{K p_i} \leq \sum_i p_i \left( \frac{1}{K p_i} - 1 \right) = 0 \quad (4.50)$$

we get the inequality

$$H(X) \leq \sum_i p_i \log_2 K = \log_2 K. \quad (4.51)$$

4-2.

- (a) From (4.12) and (4.11) we get

$$I(X, Y) = - \sum_{y \in \Omega_Y} p_X(x) \log_2(p_X(x)) + \sum_{x \in \Omega_X} \sum_{y \in \Omega_Y} p_{X,Y}(x, y) \log_2 p_{X|Y}(x|y). \quad (4.52)$$

From Bayes rule this becomes

$$I(X, Y) = - \sum_{x \in \Omega_X} p_X(x) \log_2(p_X(x)) + \sum_{x \in \Omega_X} \sum_{y \in \Omega_Y} p_{X,Y}(x, y) \log_2 \left[ \frac{p_{Y|X}(y|x) p_X(x)}{p_Y(y)} \right] \quad (4.53)$$

which reduces directly to (4.13) after algebraic manipulation.

- (b) We can show this by substituting into (4.14) in terms of the input and transition probabilities, and showing algebraically that the result is equivalent to (4.14).

4-3.

- (a) The easiest method is to use the formula

$$I(X, Y) = H(Y) - H(Y|X). \quad (4.54)$$

We want to show that the second term is independent of  $q$ . When the input is  $X = 0$ , the two outputs have probability  $p$  and  $1 - p$ , and similarly when input is  $X = 1$  the outputs have the same probabilities. Hence the entropy of the output is the same in both cases, and the conditional entropy, the average of these two entropies over the input distribution is independent of the input distribution. The result follows immediately.

- (b) We know from Figure 4-2 that  $H(Y) \leq 1$  with equality if and only if the two outputs have equal probability. Because of the symmetry of the channel, they will have equal probability when  $q = 1/2$ , and this is therefore the distribution that achieves channel capacity.

4-4. We get

$$\int_{-\infty}^{\infty} f_Y(y) \log_2 \frac{f_Y(y)}{g(y)} dy \leq 0 \quad (4.55)$$

using the inequality, which leads directly to (4.22). Equality in (4.55) holds when  $g(y) = f_Y(y)$ . Substituting a zero-mean Gaussian with variance  $\sigma^2$  for  $g(y)$ , we get (4.21). Equality holds when  $Y$  is Gaussian.

4-5. From (4.23), using the observation that  $f_{Y|X}(y|x) = f_N(y - x)$ ,

$$\begin{aligned}
 H(Y|X) &= \sum_{x \in \Omega_X} p_X(x) \int_{-\infty}^{\infty} f_N(y - x) \log_2 f_N(y - x) dy \\
 &= \sum_{x \in \Omega_X} p_X(x) \int_{-\infty}^{\infty} f_N(n) \log_2 f_N(n) dn \\
 &= \int_{-\infty}^{\infty} f_N(n) \log_2 f_N(n) dn = H(N).
 \end{aligned} \tag{4.56}$$

4-6.

$$\begin{aligned}
 I(X, Y) &= H(Y) - H(Y|X) \\
 &= - \int_{\Omega_Y} f_Y(y) \log_2 f_Y(y) dy + \sum_{x \in \Omega_X} p_X(x) \int_{\Omega_Y} f_{Y|X}(y|x) \log_2 f_{Y|X}(y|x) dy \\
 &= - \int_{\Omega_Y} \left[ f_Y(y) \log_2 f_Y(y) - \sum_{x \in \Omega_X} p_X(x) f_{Y|X}(y|x) \log_2 f_{Y|X}(y|x) \right] dy.
 \end{aligned} \tag{4.57}$$

Using

$$f_Y(y) = \sum_{x \in \Omega_X} p_X(x) f_{Y|X}(y|x), \tag{4.58}$$

we get

$$\begin{aligned}
 I(X, Y) &= \int_{\Omega_Y} \sum_{x \in \Omega_X} p_X(x) \left[ f_{Y|X}(y|x) \left[ \log_2 f_{Y|X}(y|x) - \log_2 f_Y(y) \right] \right] dy \\
 &= \sum_{x \in \Omega_X} p_X(x) \int_{\Omega_Y} f_{Y|X}(y|x) \log_2 \frac{f_{Y|X}(y|x)}{f_Y(y)} dy,
 \end{aligned} \tag{4.59}$$

from which the result follows.

4-7.

(a) Using the inequality  $\log(x) \leq (x-1)$ ,

$$\int f_Y(y) \log_2 \frac{g(y)}{f_Y(y)} dy \leq \int (g(y) - f_Y(y)) dy = 0. \tag{4.60}$$

(b) Substituting directly in,

$$\begin{aligned}
 \log_2 g(y) &= \log_2 \left\{ \prod_{n=1}^N \frac{1}{\sqrt{2\pi(\sigma_{x,n}^2 + \sigma^2)}} \exp\{-y_n^2/2(\sigma_{x,n}^2 + \sigma^2)\} \right\} \\
 &= \sum_{n=1}^N \left\{ \frac{1}{2} \log_2(2\pi(\sigma_{x,n}^2 + \sigma^2)) + \log_2 e \cdot \frac{y_n^2}{2(\sigma_{x,n}^2 + \sigma^2)} \right\}.
 \end{aligned} \tag{4.61}$$

Now,



$$\begin{aligned}
 \int f_Y(y) \log_2 g(y) dy &= \frac{1}{2} \sum_{n=1}^N \log_2(2\pi(\sigma_{x,n}^2 + \sigma^2)) + \frac{1}{2} \sum_{n=1}^N \frac{\log_2 e}{2(\sigma_{x,n}^2 + \sigma^2)} \int f_Y(y) y_n^2 dy \\
 &= \frac{1}{2} \sum_{n=1}^N \log_2(2\pi e(\sigma_{x,n}^2 + \sigma^2)).
 \end{aligned} \quad (4.62)$$

Substituting  $H(Y)$  into  $I(X, Y)$ , we get

$$I(X, Y) \leq \frac{1}{2} \sum_{n=1}^N \log_2 \left[ 1 + \frac{\sigma_{x,n}^2}{\sigma^2} \right]. \quad (4.63)$$

(c) Calculating the difference,

$$\begin{aligned}
 \frac{1}{2} \sum_{n=1}^N \log_2 \left[ 1 + \frac{\sigma_{x,n}^2}{\sigma^2} \right] - \frac{1}{2} \sum_{n=1}^N \log_2 \left[ 1 + \frac{\sigma_x^2}{N\sigma^2} \right] &= \frac{1}{2} \sum_{n=1}^N \log_2 \frac{1 + \sigma_{x,n}^2/\sigma^2}{1 + \sigma_x^2/N\sigma^2} \\
 &\leq \frac{1}{2} \sum_{n=1}^N \frac{N\sigma_{x,n}^2 - \sigma_x^2}{N\sigma^2 + \sigma_x^2} = 0,
 \end{aligned} \quad (4.64)$$

with equality if and only if  $\sigma_{x,n}^2 = \sigma_x^2/N$ .

5-1. This follows immediately since the input voltage and current of the second twoport are equal to the output voltage and current of the first twoport.

5-2. From (5.2), we get the following four equations relating the input and output voltage and current,

$$V_1 = V_+ e^{\gamma L} + V_- e^{-\gamma L} \quad (5.88)$$

$$I_1 = \frac{V_+ e^{\gamma L} - V_- e^{-\gamma L}}{Z_0} \quad (5.89)$$

$$V_2 = V_+ + V_- \quad (5.90)$$

$$I_2 = \frac{V_+ - V_-}{Z_0}. \quad (5.91)$$

Eliminating  $V_+$  and  $V_-$  from these equations, we get the result.

5-3. The direct path has distance

$$d \left[ 1 + \frac{(h_t - h_r)^2}{d^2} \right]^{1/2} \approx d + \frac{(h_t - h_r)^2}{2d}, \quad (5.92)$$

while the reflected path has distance

$$d \left[ 1 + \frac{(h_t + h_r)^2}{d^2} \right]^{1/2} \approx d + \frac{(h_t + h_r)^2}{2d}. \quad (5.93)$$

The difference in distance between the two paths is  $\Delta d = 2h_t h_r / d$ . Taking into account the reflection coefficient of  $-1$ , the superposition of the two carrier phase shifts at the receiving antenna will be

$$1 - e^{-j2kh_t h_r / d} = 1 - e^{-j4\pi h_t h_r / \lambda d} \quad (5.94)$$

and the magnitude is

$$2 \sin \left[ \frac{2\pi h_t h_r}{\lambda d} \right] \approx \frac{4\pi h_t h_r}{\lambda d} \quad (5.95)$$

- 5-4. The quadratic term is insignificant relative to the linear term if  $(vt)^2 \ll 2dvt$ , which evaluates to  $t \ll 2d/v$ . Similarly, the linear term is small relative to unity when  $2dvt \ll d^2$  which evaluates to  $t \ll d/2v$ . Thus, both conditions are essentially the same, and the result follows from the linear approximation

$$\sqrt{1 + \epsilon} \approx 1 + \epsilon/2 \quad (5.96)$$

- 5-5. First,

$$E[(\text{Re}\{R(t_0)\})^2] = \sum_m \sum_n A_m A_n E[\cos \xi_m \cos \xi_n], \quad (5.97)$$

and when  $m = n$ ,  $E[\cos^2 \xi_m] = 1/2$ , and when  $m \neq n$ ,

$$E[\cos \xi_m \cos \xi_n] = E[\cos \xi_m] E[\cos \xi_n] = 0. \quad (5.98)$$

The imaginary part is very similar. The cross-correlation is

$$E[\text{Re}\{R(t_0)\} \text{Im}\{R(t_0)\}] = \sum_m \sum_n A_m A_n E[\cos \xi_m \sin \xi_n] \quad (5.99)$$

so the only real difference is when  $m = n$ , in which case

$$E[\cos \xi_m \sin \xi_m] = 1/2 E[\cos 2\xi_m] = 0, \quad (5.100)$$

since the average is over two periods of the sinusoid.

- 5-6.

$$x(t) = 1/2 \left[ s(t) e^{j\omega_c t} + s^*(t) e^{-j\omega_c t} \right] \quad (5.101)$$

$$X(j\omega) = 1/2 \left[ S(j\omega - j\omega_c) + S^*(-j\omega - j\omega_c) \right] \quad (5.102)$$

From the bandlimited assumption,

$$X(j\omega) = \begin{cases} \frac{1}{2} S(j\omega - j\omega_c); & \omega > 0 \\ \frac{1}{2} S^*(j\omega - j\omega_c); & \omega < 0 \end{cases} \quad (5.103)$$

From (5.77),

$$Y(j\omega) = \begin{cases} \frac{1}{2} S(j\omega - j\omega_c - j\omega_0); & \omega > 0 \\ \frac{1}{2} S^*(j\omega - j\omega_c - j\omega_0); & \omega < 0 \end{cases} \quad (5.104)$$

$$= \frac{1}{2} [S(j\omega - j\omega_c - j\omega_0) + S^*(-j\omega - j\omega_c - j\omega_0)]$$

$$\begin{aligned}
 y(t) &= \frac{1}{2} [s(t)e^{j(\omega_c + \omega_0)t} s^*(t)e^{-j(\omega_c + \omega_0)t} \\
 &= \operatorname{Re}\{s(t)e^{j(\omega_c + \omega_0)t}\}
 \end{aligned} \tag{5.105}$$

- 6-1. To find  $\sigma^2$  we write the power spectrum of  $U(t)$ ,

$$S_U(j\omega) = S_N(j\omega) |F(j\omega)|^2. \tag{6.238}$$

Its power is

$$R_U(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_N(j\omega) |F(j\omega)|^2 d\omega. \tag{6.239}$$

Combining this with (6.35) we get

$$R_U(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_N(j\omega) \left| \frac{P(j\omega)}{B(j\omega)G(j\omega)} \right|^2 d\omega. \tag{6.240}$$

Since  $U(t)$  is a WSS random process, the variance of its samples equals its power, so the result follows.

- 6-2. Recalling the definition of autocorrelation for a complex-valued process,

$$\begin{aligned}
 R_Z(\tau) &= E[Z(t+\tau)Z^*(t)] = E[e^{-j\omega_c(t+\tau)}M(t+\tau)e^{j\omega_c t}M^*(t)] \\
 &= e^{-j\omega_c \tau} R_M(\tau).
 \end{aligned} \tag{6.241}$$

- 6-3. Treating only the analytic signals at input and output of the channel,

$$h(t)e^{j\omega_c t} = (g(t)e^{j\omega_c t}) * b(t) \tag{6.242}$$

and substituting an integral for the convolution,

$$h(t) = (b(t)e^{-j\omega_c t}) * g(t). \tag{6.243}$$

- 6-4.

$$h(t) * h(-t) = \int_0^T h(\tau)h(\tau-t) d\tau \tag{6.244}$$

Outside the range  $[-T, T]$  the two integrands do not overlap, likewise at the endpoints of this interval there is an overlap at only one point, and the integral will be zero.

- 6-5. Let  $g_m(t)$  and  $g_n(t)$  be orthogonal complex-valued waveforms. Calculating the cross-correlation of the passband signals, conjugating the second baseband waveform without affecting its real part,

$$\int_{-\infty}^{\infty} \sqrt{2} \operatorname{Re}\{g_m(t)e^{j\omega_c t}\} \sqrt{2} \operatorname{Re}\{g_n^*(t)e^{-j\omega_c t}\} dt. \tag{6.245}$$

Substituting for the real parts in terms of the waveforms and their conjugate, we get four cross terms in the integral. Two of those cross-terms are multiplied by  $e^{j2\omega_c t}$ , and will integrate to zero if  $\omega_c$  is large because their Fourier transforms do not overlap d.c. (the integral is just the d.c. component of their Fourier transforms). The remaining two terms evaluate to

$$\frac{1}{2} \int_{-\infty}^{\infty} (g_m(t)g_n^*(t) + g_m^*(t)g_n(t)) dt = \operatorname{Re}\left\{ \int_{-\infty}^{\infty} g_m(t)g_n^*(t) dt \right\} \tag{6.246}$$

Hence, orthogonality at baseband implies orthogonality at passband.

- 6-6. The pulses in (6.129) consist of a sinc pulse modulating a cosine. Using the tables in appendix 2-A, the Fourier transform of the sinc pulses is

$$\text{FT} \left[ \frac{1}{\sqrt{T}} \left[ \frac{\sin(\pi t/2T)}{\pi t/2T} \right] \right] = 2\sqrt{T} \text{rect}(\omega, \pi/2T). \quad (6.247)$$

Multiplying the pulse in the time domain by a cosine of frequency  $(n + 1/2)\pi/T$  will scale the Fourier transform by  $1/2$  and shift it up and down in frequency by  $(n + 1/2)\pi/T$ . The frequency-domain plots in Figure 6-38 result. These plots make it clear that

$$H_n(j\omega)H_l^*(j\omega) = \sqrt{T}H_n(j\omega)\delta_{l-n}. \quad (6.248)$$

Thus, for any  $n \neq l$ , the sum in (6.128) is zero. So it remains to be shown only that when  $n = l$ , the sum is constant, or using (6.248),

$$\frac{1}{\sqrt{T}} \sum_{m=-\infty}^{\infty} H_n(j(\omega + m\frac{2\pi}{T})) = 1. \quad (6.249)$$

Equivalently, we require that  $\sqrt{T}H_n(j\omega)$  satisfy the Nyquist criterion (6.23), or that  $\sqrt{T}h_n(t)$  satisfy the zero-forcing criterion (6.21). From (6.129),

$$\sqrt{T}h_n(kT) = \delta_k, \quad (6.250)$$

verifying that (6.21) is satisfied.

- 6-7. To see this, note first that

$$\begin{aligned} h_n(t)h_n(t-kT) &= \frac{1}{2}q(t)q(t-kT) \left[ \cos[(n + 1/2)k\pi] + \cos[(2n + 1)\pi t/T - (n + 1/2)k\pi] \right]. \end{aligned} \quad (6.251)$$

Since  $q(t)$  is bandlimited to  $\pi/T$ ,  $q(t)q(t-kT)$  is bandlimited to  $2\pi/T$ , and thus when it is multiplied by a cosine with frequency  $(2n + 1)\pi/T \geq 3\pi/T$ , the spectrum does not overlap d.c. This term must therefore integrate to zero, so

$$\int_{-\infty}^{\infty} h_n(t)h_n(t-kT) dt = \frac{1}{2} \cos[(n + 1/2)k\pi] \int_{-\infty}^{\infty} q(t)q(t-kT) dt. \quad (6.253)$$

For  $k = 0$ , using (6.131), this integral has value unity. For  $k$  even but not zero, (6.131) implies that the integral has value zero. For  $k$  odd, the integral is zero because  $\cos((n + 1/2)k\pi)$  is zero. Hence, (6.125) holds.

- 6-8. Writing out the convolution in (6.127), we need to show that

$$\frac{2}{T} \int_0^T \sin(\omega_n \tau) \sin(\omega_l(\tau - kT)) w(\tau - kT) d\tau = \delta_k \delta_{l-n}, \quad (6.254)$$

for  $l, n = 0$ , or 1, and for any integer  $k$ . When  $k \neq 0$ ,  $w(\tau - kT) = 0$  within the range of the integral, so it will suffice to show that

$$\frac{2}{T} \int_0^T \sin(\omega_n \tau) \sin(\omega_l \tau) d\tau = \delta_{l-n}, \quad (6.255)$$

or equivalently, that

$$\frac{1}{T} \int_0^T [\cos((\omega_n - \omega_l)\tau) - \cos((\omega_n + \omega_l)\tau)] d\tau = \delta_{l-n}. \quad (6.256)$$

Using (6.144), we need to show that

$$\frac{1}{T} \int_0^T [\cos((M_n - M_l)2\pi\tau/T) - \cos((M_n + M_l)2\pi\tau/T)] d\tau = \delta_{l-n} . \quad (6.257)$$

When  $n \neq l$ , both terms under the integral are cosines that are integrated over an integer number of cycles, and hence integrate to zero. When  $n = l$ , the second term integrates to zero, but the first term integrates to  $T$ , thus establishing the result.

- 6-9. Evaluate (6.152) at the boundary between two symbols,  $t = kT$ , and find that for the phase to be the same on either side of the boundary we need

$$\omega_c kT + b_k \frac{\pi kT}{2T} + \phi_k = \omega_c kT + b_{k-1} \frac{\pi kT}{2T} + \phi_{k-1} , \quad (6.258)$$

which easily reduces to the desired form.

- 6-10. The sampled receive filter output is

$$\begin{aligned} g_0(t) * f(t) \Big|_{t=0} &= \frac{2}{T} \int_0^T \sin(\omega_0\tau) \sin(\omega_0\tau + \theta) d\tau \\ &= \frac{1}{T} \int_0^T [\cos(\theta) - \cos(2\omega_0\tau + \theta)] d\tau . \end{aligned} \quad (6.259)$$

The second term in the integral integrates to zero for any fixed  $\theta$ , and the first term is constant, so

$$g_0(t) * f(t) \Big|_{t=0} = \cos(\theta) . \quad (6.260)$$

- 6-11.

$$\int_{-\infty}^{\infty} g_n(t) g_k(t) dt = \frac{1}{T} \int_0^T e^{j\omega_n t} e^{-j\omega_k t} dt = \frac{1}{T} \int_0^T e^{j2\pi(n-k)T} . \quad (6.261)$$

When  $n = k$ , this clearly equals unity. When  $n \neq k$ , we are integrating over an integer number of cycles of a complex exponential, getting zero.

- 6-12. To show that  $Z(t)$  is WSS, we must first show that  $E[Z(t)]$  is independent of  $t$ . Evaluating this mean value,

$$E[Z(t)] = e^{j\omega_c t} E[S(t)] = 0 \quad (6.262)$$

because of the zero-mean property of  $S(t)$ . Next we need to evaluate the autocorrelation  $R_Z(\tau)$ ,

$$\begin{aligned} R_Z(\tau) &= E\{Z(t+\tau)Z^*(t)\} \\ &= E\{e^{j\omega_c(t+\tau)} e^{-j\omega_c t} S(t+\tau) S^*(t)\} \\ &= e^{j\omega_c \tau} R_S(\tau) . \end{aligned} \quad (6.263)$$

Since the result is independent of  $t$ , we have wide-sense stationarity.

- 6-13. Evaluating the cross-correlation of  $Z(t)$  and  $Z^*(t)$  with delay  $\tau$ ,

$$E[Z(t+\tau)Z^*(t)] = e^{j\omega_c(2t+\tau)} R_{SS^*}(\tau) . \quad (6.264)$$

Clearly this quantity is independent of  $t$  for all  $\tau$  if and only if  $R_{SS^*}(\tau) = 0$ . If that is the case, then (6.264) is the cross-correlation function  $R_{ZZ^*}(\tau)$ , and  $R_{ZZ^*}(\tau) = 0$ .

6-14. Evaluating the autocorrelation explicitly,

$$R_{SS^*}(\tau) = E[S(t + \tau)S^*(t)] = R_R(\tau) - R_I(\tau) + j(R_{RI}(\tau) + R_{IR}(\tau)) \quad (6.265)$$

This is clearly zero for all  $\tau$  only under the conditions stated. Also, since  $R_{IR}(\tau) = R_{RI}(-\tau)$ , it follows that  $R_{RI}(\tau) = -R_{RI}(-\tau)$  must also be satisfied.

6-15. The cross-correlation of  $S(t)$  and  $S^*(t)$  is

$$E[S(t + \tau)S^*(t)] = \frac{1}{T} \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} E[A_k A_m] E[h(t + \tau - kT + \Theta)h^*(t - mT + \Theta)] \quad (6.266)$$

Clearly this quantity is zero if  $E[A_k A_m] = 0$ . To find sufficient conditions for this, let

$$A_k = I_k + jQ_k \quad (6.267)$$

and calculate

$$E[A_k A_m] = E[I_k I_m] - E[Q_k Q_m] + j(E[I_k Q_m] + E[Q_k I_m]) \quad (6.268)$$

Clearly for (6.268) to be zero it is sufficient that

$$E[I_k I_m] = E[Q_k Q_m] \quad (6.269)$$

and

$$E[I_k Q_m] = 0 \quad (6.270)$$

for all  $k$  and  $m$ .

6-16. From the definition of the autocorrelation,

$$\begin{aligned} R_X(\tau) &= 0.5E[(Z(t + \tau) + Z^*(t + \tau))(Z^*(t) + Z(t))] \\ &= 0.5(R_Z(\tau) + R_Z^*(\tau) + R_{ZZ^*}(\tau) + R_{ZZ^*}^*(\tau)) \end{aligned} \quad (6.271)$$

When (6.186) is satisfied,  $R_X(\tau)$  in (6.271) is not a function of  $t$  and  $X(t)$  is WSS. Further, under these conditions (6.187) is satisfied and

$$R_X(\tau) = 0.5(R_Z(\tau) + R_Z^*(\tau)) \quad (6.272)$$

which reduces to (6.194). In (6.272) we have assumed that  $Z(t)$  is WSS, which we saw earlier requires that  $S(t)$  have zero mean and be WSS.

6-17. If  $X(t)$  is WSS, its autocorrelation is given by (6.194). From appendix 2-A,

$$S_X(j\omega) = [S_Z(j\omega)]_e \quad (6.273)$$

where

$$[S_Z(j\omega)]_e = 0.5[S_Z(j\omega) + S_Z^*(-j\omega)]. \quad (6.274)$$

Since  $S_Z(j\omega)$  is real,

$$S_X(j\omega) = 0.5[S_Z(j\omega) + S_Z(-j\omega)]. \quad (6.275)$$

Substituting from (6.185), the result follows.

6-18. We have

$$R_S(\tau) = E[S(t + \tau)S^*(t)] = R_R(\tau) + R_I(\tau) + j(R_{IR}(\tau) - R_{RI}(\tau)) \quad (6.276)$$

Employing the results of exercise 6-14, this simplifies to

$$R_S(\tau) = 2R_R(\tau) - 2j R_{RI}(\tau) \quad (6.277)$$

6-19. Using (6.195) and (3.59) we write

$$\begin{aligned} R_X(0) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(j\omega) d\omega \\ &= \frac{1}{4\pi} \left[ \int_{-\infty}^{\infty} S_S(j\omega - j\omega_c) d\omega + \int_{-\infty}^{\infty} S_S(-j\omega - j\omega_c) d\omega \right]. \end{aligned} \quad (6.278)$$

Changing variables in each integral

$$R_X(0) = 0.5[R_S(0) + R_S(0)] = R_S(0). \quad (6.279)$$

8-1. Writing the  $2\text{-Re}\{\}$  as the sum of the argument plus its conjugate, the right-hand side is

$$\int_{-\infty}^{\infty} (s_l(t)\phi_n^*(t) + s_l^*(t)\phi_n(t)e^{-j2\omega_c t}) dt. \quad (8.198)$$

The second term must be zero, since  $s_l(t)$  and  $\phi_n(t)$  are both bandlimited to  $\omega_c$ , and hence their product is bandlimited to  $2\omega_c$ . Thus, we are left with

$$\int_{-\infty}^{\infty} s_l(t)\phi_n^*(t) dt = S_{l,n} \quad (8.199)$$

from (8.75).

8-2. Letting  $x = N(M^{1/N} - 1)$  where  $N = BT$ , we get

$$M = (1 + \frac{x}{N})^N \rightarrow e^x \text{ as } N \rightarrow \infty. \quad (8.200)$$

Hence  $x \rightarrow \log_e M$  as  $N \rightarrow \infty$ .

8-3. The Chernoff bound is

$$1 - F_X(x) \leq e^{x-a-x \ln(x/a)}, \quad (8.201)$$

$$F_X(x) \leq e^{x-a-x \ln(x/a)}. \quad (8.202)$$

Substituting  $a + \delta$  for  $x$  in (8.201), we get for the exponent,

$$\delta - (a + \delta) \ln(1 + \frac{\delta}{a}) \approx \delta - (a + \delta)(\frac{\delta}{a} - 0.5(\frac{\delta}{a})^2) \approx -\frac{\delta^2}{2a} \quad (8.203)$$

where we have eliminated the term in  $\delta^3$ . Similarly, substituting  $a - \delta$  for  $x$  in (8.202), we get the same answer.

8-4. We can write

$$x(t) = \text{Re}\{Be^{j\omega_1 t} \pm Ae^{j(\omega_1 + \omega_p)t}\} = \text{Re}\{(B \pm Ae^{j\omega_p t})e^{j\omega_1 t}\}. \quad (8.204)$$

The envelope of the signal is

$$E^2(t) = |B \pm Ae^{j\omega_p t}|^2 \quad (8.205)$$

as given by (8.183).

8-5. Plugging (8.189) into (8.190) we get

$$\begin{aligned} Q &= \int_0^T \pm 2AB \cos^2(\omega_{\text{IF}} t) dt + \int_0^T N(t) \cos(\omega_{\text{IF}} t) dt \\ &= \int_0^T \pm 2AB \left[ 0.5 + 0.5 \cos(2\omega_{\text{IF}} t) \right] dt + \int_0^T N(t) \cos(\omega_{\text{IF}} t) dt. \end{aligned} \quad (8.206)$$

If either  $\omega_{\text{IF}}$  is large or  $\omega_{\text{IF}} T = K 2\pi$  where  $K$  is an integer then the double frequency term can be neglected, getting

$$Q = \pm ABT + N \quad (8.207)$$

where

$$N = \int_0^T N(t) \cos(\omega_{\text{IF}} t) dt. \quad (8.208)$$

The desired signal has energy (per symbol)  $(ABT)^2$  and the noise term  $N$  has zero mean and variance given by

$$\begin{aligned} E[N^2] &= \int_0^T \int_0^T E[N(t)N(\tau)] \cos(\omega_{\text{IF}} t) \cos(\omega_{\text{IF}} \tau) dt d\tau \\ &= \int_0^T \int_0^T B^2 \delta(t - \tau) \cos(\omega_{\text{IF}} t) \cos(\omega_{\text{IF}} \tau) dt d\tau \\ &= \int_0^T B^2 \cos^2(\omega_{\text{IF}} t) dt \\ &= B^2 T / 2. \end{aligned} \quad (8.209)$$

The SNR follows.

$$\frac{1}{A_{h,n}^2 \cdot G_{h,n}^* (1/z^*)}$$

9-1. Writing  $z = Ae^{j\phi}$ , the integral on the right hand side is

$$\int_{-\pi}^{\pi} \exp\{ \operatorname{Re}\{ Ae^{j(\theta-\phi)} \} \} d\theta = \int_{-\pi}^{\pi} \exp\{ A \cos(\theta - \phi) \} d\theta. \quad (9.173)$$

Since the integral is over one period of the periodic integrand, the value of the integral is independent of  $\phi$ , and hence the result follows for  $\phi = 0$ .

9-2. This is easily done by considering each of the four possible paths of length two through the trellis and computing the distances of the shortest error events. It is easy to argue that all longer error events have greater distances.

9-3. From Bayes' rule we can write

$$p(\hat{\Psi}) = p(\hat{\Psi}_K | \hat{\Psi}_{K-1}, \dots, \hat{\Psi}_0) p(\hat{\Psi}_{K-1}, \dots, \hat{\Psi}_0). \quad (9.174)$$

From the Markov property this becomes

$$p(\hat{\Psi}) = p(\hat{\Psi}_K | \hat{\Psi}_{K-1}) p(\hat{\Psi}_{K-1}, \dots, \hat{\Psi}_0). \quad (9.175)$$

Repeat with the second factor, and iterate until the desired form results.



$$\frac{S_A}{S_A H H^* + S_Z},$$

10-1. A plot of  $X_k$  as a function of  $A_k$ , for a given  $v_k$ , is shown in Figure 10-29. A given threshold  $x$  is shown, the probability that  $X_k \leq x$  corresponds to the range of  $A_k$  that is shaded. Assuming that  $A_k$  is uniformly distributed, then  $\Pr\{X_k \leq x\}$  is easily seen to be proportional to  $x$ , regardless of the value of  $v_k$ . Hence  $X_k$  is uniformly distributed.

10-2. By straightforward manipulation,

$$\begin{aligned} S_E &= S_Y |C|^2 - S_A (H C + H^* C^*) + S_A \\ &= S_Y |C - S_A S_Y^{-1} H^*|^2 + S_A - S_A^2 S_Y^{-1} |H|^2 \end{aligned} \quad (10.176)$$

and from there it is just some minor algebra.

10-3. The Fourier transform of the input pulse is  $e^{j\omega\alpha_0} H(j\omega)$ , and hence at the output of the matched filter the pulse has Fourier transform  $e^{j\omega\alpha_0} |H(j\omega)|^2$ . After sampling we get the result of (10.97).

10-4. Assuming less than 100% excess bandwidth, a sampling rate of  $2/T$  suffices to represent the matched filter impulse response,

$$H(j\omega) = \sum_{m=-\infty}^{\infty} h_m e^{-j\omega m T/2}, \quad (10.177)$$

and the matched filter can be represented in discrete time as a filter with impulse response  $h_{-m}^*$ . When we double the sampling rate in  $C(z)$ , yielding a new filter with coefficients  $c_k'$ , we have

$$c_k' = \begin{cases} c_{k/2}, & k \text{ even} \\ 0, & k \text{ odd} \end{cases} \quad (10.178)$$

The FSE has impulse response

$$f_k = \sum_{m=-\infty}^{\infty} c_m' h_{m-k}^* = \sum_{r=-\infty}^{\infty} c_{2r}' h_{2r-k}^* = \sum_{r=-\infty}^{\infty} c_r h_{2r-k}^* \quad (10.179)$$

and the transfer function of this filter is

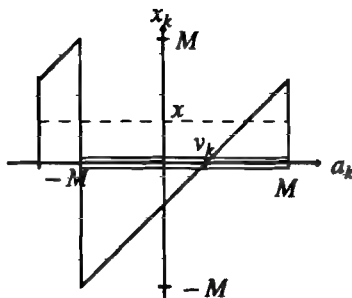


Figure 10-29. A plot of the nonlinear relationship between  $a_k$  and  $x_k$  from (10.47) for a given  $v_k$ .

$$\begin{aligned}
 \sum_{k=-\infty}^{\infty} f_k e^{-j\omega kT/2} &= \sum_{k=-\infty}^{\infty} e^{-j\omega kT/2} \sum_{r=-\infty}^{\infty} c_r h_{2r-k}^* \\
 &= \sum_{r=-\infty}^{\infty} c_r \sum_{k=-\infty}^{\infty} h_{2r-k}^* e^{-j\omega kT/2}
 \end{aligned} \quad (10.180)$$

and letting  $m = 2r - k$  this becomes

$$\begin{aligned}
 \sum_{r=-\infty}^{\infty} c_r \sum_{m=-\infty}^{\infty} h_m^* e^{-j\omega(2r-m)T/2} &= \sum_{r=-\infty}^{\infty} c_r e^{-j\omega rT} \sum_{m=-\infty}^{\infty} h_m^* e^{-j\omega mT/2} \\
 &= C(e^{j\omega T}) H^*(j\omega) .
 \end{aligned} \quad (10.181)$$

10-5. Consider Figure 10-18b, we will show it is equivalent to Figure 10-18a. Since the transfer function  $C_1(e^{j\omega T})$  is periodic in frequency, it can be expanded in a Fourier series,

$$C_1(e^{j\omega T}) = \sum_m c_m e^{-j\omega mT} \quad (10.182)$$

and hence it has impulse response

$$c_1(t) = \sum_m c_m \delta(t - mT). \quad (10.183)$$

If the matched filter output is called  $x(t)$  and the output of the filter  $C_1(e^{j\omega T})$  is called  $y(t)$ , then from (10.183)

$$y(t) = \sum_m c_m x(t - mT) \quad (10.184)$$

and therefore

$$Q_k = y(kT) = \sum_m c_m x((k-m)T) , \quad (10.185)$$

which is evidently Figure 10-18a.

10-6. We get

$$\begin{aligned}
 \Pr\{W_k \neq 0\} &= 0.5(\Pr\{W_k = 2\} + \Pr\{W_k = -2\}) \\
 &= 0.5(\Pr\{Z_k < -1 - V_k\} + \Pr\{Z_k > 1 - V_k\}) \\
 &= 0.5(1 - \Pr\{-1 - V_k \leq Z_k \leq 1 - V_k\}) \leq 1/2 .
 \end{aligned} \quad (10.186)$$

11-1. Write

$$\begin{aligned}
 |E_k|^2 &= E_k^* E_k \\
 &= (A_k^* - \mathbf{c}^{*'} \mathbf{r}_k^*)(A_k - \mathbf{r}_k' \mathbf{c}) \\
 &= |A_k|^2 - \mathbf{c}^{*'} \mathbf{r}_k^* A_k - \mathbf{r}_k' \mathbf{c} A_k^* + \mathbf{c}^{*'} \mathbf{r}_k^* \mathbf{r}_k' \mathbf{c} .
 \end{aligned} \quad (11.124)$$

Taking the expected value, we get the stated result.

11-2.  $R_k$  is the sum of a signal and a noise term, which are uncorrelated and can thus be considered directly. From exercise 3-11, the autocorrelation of the signal component is

$$p_k * p_{-k}^* * \sigma_a^2 \delta_k = \sigma_a^2 p_k * p_{-k}^* . \quad (11.125)$$

Similarly the noise component can be determined directly from 8-mex\_nac,

$$E[R_{k+j} R_k^*] = 2N_0 \int_{-\infty}^{\infty} f^*(t - (k+j)T) e^{-j\omega_c t} f(t - kT) e^{j\omega_c t} dt = 2N_0 \rho_f(j) . \quad (11.126)$$

11-3.

(a) By the definition of  $\Phi$ ,

$$\begin{aligned}\Phi^{*'} &= E\{r_k^* r_k'\}^{*'} \\ &= E\{r_k r_k^{*'}\}^* = E\{r_k^* r_k'\} \\ &= \Phi.\end{aligned}\quad (11.127)$$

(b) The Toeplitz property follows directly from the assumption that the  $R_k$  input random process is wide-sense stationary.

(c) The positive semidefinite property follows from

$$\mathbf{x}^{*'} \Phi \mathbf{x} = E\{|\mathbf{r}_k' \mathbf{x}|^2\} \geq 0. \quad (11.128)$$

11-4. The easiest method is to note that if the MSE reduces to a Hermitian form, the matrix must be  $\Phi$ . Hence, we assume the form of the result,

$$E\{|E_k|^2\} = a + (\mathbf{c} - \mathbf{c}_{\text{opt}})^{*'} \Phi (\mathbf{c} - \mathbf{c}_{\text{opt}}) \quad (11.129)$$

for unknown constants  $a$  and  $\mathbf{c}_{\text{opt}}$ , and multiply out and equate terms to determine the constants.

11-5.

(a) From the Hermitian property we get

$$(\Phi_R + j\Phi_I)^{*'} = \Phi_R + j\Phi_I \quad (11.130)$$

and the result follows from equating real and imaginary parts.

(b) We have that

$$\mathbf{c}^{*'} \Phi \mathbf{c} = (\mathbf{c}_R' - j\mathbf{c}_I')(\Phi_R + j\Phi_I)(\mathbf{c}_R + j\mathbf{c}_I) \quad (11.131)$$

and multiplying out and taking the real part (note we don't need to bother with the imaginary part at all since we know in advance it is zero),

$$\mathbf{c}^{*'} \Phi \mathbf{c} = \mathbf{c}_R' \Phi_R \mathbf{c}_R + \mathbf{c}_I' \Phi_R \mathbf{c}_I - 2\mathbf{c}_R' \Phi_I \mathbf{c}_I \quad (11.132)$$

or equivalently

$$\mathbf{c}^{*'} \Phi \mathbf{c} = \mathbf{c}_R' \Phi_R \mathbf{c}_R + \mathbf{c}_I' \Phi_R \mathbf{c}_I + 2\mathbf{c}_I' \Phi_I \mathbf{c}_R. \quad (11.133)$$

Taking the gradients, we get the desired results. Similarly,

$$\text{Re}\{\mathbf{c}^{*'} \alpha\} = \text{Re}(\mathbf{c}_R - j\mathbf{c}_I)'(\alpha_R + j\alpha_I) = \mathbf{c}_R' \alpha_R + \mathbf{c}_I' \alpha_I. \quad (11.134)$$

(c) By the given definition of (11.23),

$$\nabla_{\mathbf{c}} \mathbf{c}^{*'} \Phi \mathbf{c} = 2(\Phi_R \mathbf{c}_R - \Phi_I \mathbf{c}_I) + j2(\Phi_R \mathbf{c}_I + \Phi_I \mathbf{c}_R) \quad (11.135)$$

which is the same as

$$\nabla_{\mathbf{c}} \mathbf{c}^{*'} \Phi \mathbf{c} = 2(\Phi_R + j\Phi_I)(\mathbf{c}_R + j\Phi_I). \quad (11.136)$$

Similarly,

$$\nabla_{\mathbf{c}} \text{Re}\{\mathbf{c}^{*'} \alpha\} = \alpha_R + j\alpha_I = \alpha. \quad (11.137)$$

11-6.

- (a) From the eigenvalue equation

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \quad (11.138)$$

where  $\lambda$  is an eigenvalue and  $\mathbf{v}$  is an associated eigenvector, we can take the conjugate and transpose to get

$$\mathbf{v}^{*'}\mathbf{A}^{*'} = \mathbf{v}^{*'}\mathbf{A} = \lambda^*\mathbf{v}^{*'} \quad (11.139)$$

Premultiplying and postmultiplying the two forms by appropriate vectors, we get

$$\mathbf{v}^{*'}\mathbf{A}\mathbf{v} = \lambda\mathbf{v}^{*'}\mathbf{v} \quad (11.140)$$

$$\mathbf{v}^{*'}\mathbf{A}\mathbf{v} = \lambda^*\mathbf{v}^{*'}\mathbf{v} \quad (11.141)$$

and subtracting these two equations

$$0 = (\lambda - \lambda^*)(\mathbf{v}^{*'}\mathbf{v}). \quad (11.142)$$

Since the second term is a vector norm and is therefore positive for  $\mathbf{v} \neq 0$ , it follows that  $\lambda = \lambda^*$  and  $\lambda$  is real.

- (b) Assume that
- $\lambda_1 \neq \lambda_2$
- are eigenvalues of
- $\mathbf{A}$
- . Then we get

$$\mathbf{A}\mathbf{v}_1 = \lambda_1\mathbf{v}_1 \quad (11.143)$$

$$\mathbf{A}\mathbf{v}_2 = \lambda_2\mathbf{v}_2 \quad (11.144)$$

and taking the conjugate-transpose and premultiplying and postmultiplying by the appropriate vector we get

$$\mathbf{v}_1^{*'}\mathbf{A}\mathbf{v}_2 = \lambda_1\mathbf{v}_1^{*'}\mathbf{v}_2 \quad (11.145)$$

$$\mathbf{v}_1^{*'}\mathbf{A}\mathbf{v}_2 = \lambda_2\mathbf{v}_1^{*'}\mathbf{v}_2. \quad (11.146)$$

Subtracting these equations,

$$0 = (\lambda_1 - \lambda_2)(\mathbf{v}_1^{*'}\mathbf{v}_2) \quad (11.147)$$

which implies that  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are orthogonal.

- (c) From the definition of the modal matrix,

$$\mathbf{V}^{*'}\mathbf{V} = [\mathbf{v}_1^{*'} \cdots \mathbf{v}_N^{*'}][\mathbf{v}_1 \cdots \mathbf{v}_N] = \mathbf{I} \quad (11.148)$$

from part b.

- (d) This follows from replacing the modal matrix by its constituent eigenvectors and multiplying out.
- (e) Let  $\mathbf{v}$  be an eigenvector and  $\lambda$  be an associated eigenvalue, then we get

$$\lambda = \frac{\mathbf{v}^{*'}\mathbf{A}\mathbf{v}}{\mathbf{v}^{*'}\mathbf{v}} \quad (11.149)$$

which is non-negative by assumption.

11-7. This follows directly from multiplying out the diagonalizing transformation of (11.35).

11-8. Noting that

$$(\mathbf{I} - \beta\Phi)\mathbf{v}_i = \mathbf{v}_i - \beta\Phi\mathbf{v}_i = \mathbf{v}_i - \beta\lambda_i\mathbf{v}_i = (1 - \beta\lambda_i)\mathbf{v}_i \quad (11.150)$$

it follows that  $\mathbf{v}_i$  is an eigenvector and  $(1 - \beta\lambda_i)$  is an eigenvalue of  $(\mathbf{I} - \beta\Phi)$ .

11-9. By straightforward manipulations,

$$\begin{aligned} \mathbf{x}^* \Phi \mathbf{x} &= \mathbf{x}^* \left( \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{v}_i^* \right) \mathbf{x} \\ &= \sum_{i=1}^n \lambda_i \mathbf{x}^* \mathbf{v}_i \mathbf{v}_i^* \mathbf{x} \\ &= \sum_{i=1}^n \lambda_i |\mathbf{x}^* \mathbf{v}_i|^2 \end{aligned} \quad (11.151)$$

11-10. Substituting into (11.53),

$$\mathbf{q}_{k+1} = \Gamma_k \mathbf{q}_k + (\Gamma_k - \mathbf{I})\mathbf{c}_{\text{opt}} + \beta A_k \mathbf{r}_k^* \quad (11.152)$$

Simplifying the term  $(\Gamma_k - \mathbf{I})$ , we get

$$\mathbf{q}_{k+1} = \Gamma_k \mathbf{q}_k + \beta(A_k - \mathbf{r}_k^* \mathbf{c}_{\text{opt}}) \mathbf{r}_k^* \quad (11.153)$$

which is the desired result.

11-11. The causal coefficients will be different because the input is data symbols rather than the data symbols filtered by the channel and receive filter impulse responses. The non-causal coefficients will be different because the causal coefficients do not cause noise enhancement, and hence there is more freedom in the choice of non-causal coefficients.

11-12. The error is given by

$$E_k = A_k - \mathbf{c}' \mathbf{w}_k \quad (11.154)$$

where

$$\mathbf{w}_k^* = [w_{-QN-1} \cdots w_0] \quad (11.155)$$

and

$$\mathbf{w}_k = \sum_{l \in I} p_l A_{k-l} \quad (11.156)$$

where the index set  $I$  consists of all integers except  $[1, M]$ . The value of  $E[|E_k|^2]$  is identical to (11.8) except that  $\Phi$  and  $\alpha$  are

$$\Phi = E[\mathbf{w}_k^* \mathbf{w}_k^*] \quad (11.157)$$

$$\alpha = E[A_k \mathbf{w}_k^*]. \quad (11.158)$$

All that remains is to determine the elements of this matrix and vector. We have

$$\begin{aligned} E[\mathbf{w}_{k+i}^* \mathbf{w}_{k+j}] &= E\left[\sum_{l \in I} p_{l-i}^* A_{k-l}^* \sum_{m \in I} p_{m-j} A_{k+j-m}\right] \\ &= \sigma_a^2 \sum_{l \in I} p_{l-i}^* p_{l+j-n}. \end{aligned} \quad (11.159)$$

Similarly,

$$E[A_k \sum_{l \in I} p_{l-i}^* A_{k-l}^*] = p_i. \quad (11.160)$$

- 11-13. First, let's ignore  $f(t)$ , and later replace  $b(t)$  by  $b(t) * f(t)$ . Secondly, let's eliminate the real-part and phase splitter by dealing only with the analytic signal. Then the output of the channel is

$$\begin{aligned} e^{-j\omega_c t} \int_{-\infty}^{\infty} b(\tau) \sum_k A_k g(t-\tau-kT) e^{j\omega_c(t-\tau)} d\tau \\ = e^{j\Delta\omega t} \sum_k A_k \int_{-\infty}^{\infty} b(\tau) e^{-j\omega_c \tau} g(t-kT-\tau) d\tau. \end{aligned} \quad (11.161)$$

- 11-14. By direct calculation,

$$E[\tilde{A}_m \tilde{A}_n^*] = e^{j\Delta\omega(m-n)T} E[A_m A_n^*]. \quad (11.162)$$

Hence, the rotated symbols are wide-sense stationary and the power spectrum relation follows directly.

- 11-15. Let  $R_k = c + jd$  and write

$$E[|c + jd|^4] = E[c^4 + d^4 + 2c^2 d^2] \quad (11.163)$$

and using the fact that

$$E[c^2] = E[d^2] = \frac{1}{2}\phi_0 \quad (11.164)$$

$$E[c^4] = E[d^4] = 3 \cdot (\frac{1}{2}\phi_0)^2 \quad (11.165)$$

we get (11.107).

- 12-1. We must find the response of the cutoff filter  $\frac{1}{\tau} e^{-\frac{t}{\tau}}$  to the biphas pulse at time  $t = T$ . This is

$$\begin{aligned} \frac{1}{\tau} \int_0^T g(u) e^{-\frac{(T-u)}{\tau}} du &= \frac{1}{\tau} e^{-\frac{T}{\tau}} \left\{ \int_0^{\frac{T}{2}} e^{\frac{u}{\tau}} du - \int_{\frac{T}{2}}^T e^{\frac{u}{\tau}} du \right\} \\ &= -(1 - e^{-\frac{T}{2\tau}})^2. \end{aligned} \quad (12.96)$$

We can relate this to  $\beta$  by noting that

$$\beta = \frac{1}{2\pi} \frac{T}{\tau}. \quad (12.97)$$

- 12-2.

$$|1 - e^{-j\omega 2T}|^2 = 4\sin^2 \omega T \quad (12.98)$$

which has a zero at  $\omega T = \pi$  or half the symbol rate.

- 12-3. The RDS of two adjacent symbols are independent of one another. Hence, the RDS at any point in time can be thought of as the sum of two RDS's, one for even symbols and one for odd symbols. The DSV is four, and hence the DSV is made four times as large.

- 12-4. The coder is represented by the following equations,

$$c_k = b_k \oplus c_{k-1} \quad (12.99)$$

$$a_k = c_k - c_{k-1} = (b_k \oplus c_{k-1}) - c_{k-1} \quad (12.100)$$

$$s_k = s_{k-1} + a_k. \quad (12.101)$$

Plugging into (12.101) we can get the following truth table:

$b_k$	$c_{k-1}$	$a_k$	$s_k$
0	0	0	$s_{k-1}$
0	1	0	$s_{k-1}$
1	0	+	$s_{k-1} + 1$
1	1	-	$s_{k-1} - 1$

We see that  $b_k = 0$  is always mapped into  $a_k = 0$ . Since  $s_k = s_{k-1} + 1$  implies that  $s_{k-1} = 0$ , and  $s_k = s_{k-1} - 1$  implies that  $s_{k-1} = 1$ , we get the desired result.

- 12-5. Assuming we start at the next bit immediately following a stuffed bit, Problem 3-13 predicts that the average number of bits before the next run of  $k$  "zeros" is

$$f_k = \frac{1 - q^k}{(1 - q)q^k}. \quad (12.102)$$

On average, then, this is the number of information bits between stuffed bits. The factor by which the bit rate is increased is therefore

$$\frac{1 + f_k}{f_k} = 1 + \frac{1}{f_k}. \quad (12.103)$$

- 12-6. For dicode, the precoder is specified by  $c_{k+1} = c_k \oplus b_{k+1}$ , and iterating

$$c_{k+N} = c_k \oplus b_{k+1} \oplus \cdots \oplus b_{k+N}. \quad (12.104)$$

The first and second terms,  $c_k$  and  $b_{k+1} \oplus \cdots \oplus b_{k+N}$ , are independent because of the presumed independence of the  $b_k$  and the fact that  $c_k$  is a function of  $b_k, b_{k-1}, \dots$ . It is then simple to verify that  $\Pr\{c_{k+N} | c_k\} = 1/2$ , and since this quantity is independent of  $c_k$  we have proven the desired independence. The modified duobinary case, which consists of two interleaved dicode precoders, follows immediately.

For the case of duobinary, the precoder is specified by  $c_{k+1} = c_k \oplus b_{k+1} \oplus 1$ , and hence

$$c_{k+N} = \begin{cases} c_k \oplus b_{k+1} \oplus \cdots \oplus b_{k+N}, & N \text{ even} \\ c_k \oplus b_{k+1} \oplus \cdots \oplus b_{k+N} \oplus 1, & N \text{ odd} \end{cases} \quad (12.105)$$

Again it is easy to verify that  $\Pr\{c_{k+N} | c_k\} = 1/2$  using the known independence.

- 12-7. The extra  $\pi/2$  in (12.57) means that we can rewrite it

$$X(t) = \sin \left[ \omega_c t + \pi \int_{-\infty}^t S(\tau) d\tau \right]. \quad (12.106)$$

Since  $g(t)$  is constant over  $0 \leq t < T$  we can write

$$X(t) = \sin \left[ \omega_c t + \pi \int_{-\infty}^{mT} S(\tau) d\tau + \pi \int_{mT}^t S(\tau) d\tau \right]. \quad (12.107)$$

In the interval  $mT \leq t < (m+1)T$  the second integral becomes

$$\pi \int_{mT}^t S(\tau) d\tau = \frac{\pi}{2T} (t - mT) A_m \quad (12.108)$$

so we identify  $b_m = A_m$  and

$$\phi_m = \pi \int_{-\infty}^{mT} S(\tau) d\tau - \frac{\pi}{2} m A_m. \quad (12.109)$$

To show that (6.153) is satisfied, verify by simple subtraction that

$$\phi_m - \phi_{m-1} = (A_{m-1} - A_m) \pi m / 2. \quad (12.110)$$

### 12-8.

- (a) By direct calculation of the autocorrelation of the scrambled sequence,

$$R(k, l) = E[A_k s_k A_{k+l} s_{k+l}] = s_k s_{k+l} R_a(l). \quad (12.111)$$

Thus, the spectrum is cyclostationary with period  $r$  since  $s_k s_{k+l}$  is periodic in  $k$  with period  $r$ .

- (b) Averaging the autocorrelation over one period, we get

$$\begin{aligned} R(l) &= \frac{1}{r} \left( \sum_{k=0}^{r-1} s_k s_{k+l} \right) R_a(l) \\ &= R_s(l) R_a(l) \\ &= \begin{cases} R_a(l), & l = g_{h,k}, \text{ } l \text{ an integer} \\ -\frac{1}{r} R_a(l), & l \neq g_{h,k} \end{cases} \end{aligned} \quad (12.112)$$

Assuming  $R_a(l) \rightarrow 0$  as  $l \rightarrow \infty$ ,  $R_a(g_{h,k}) \rightarrow 0$  as  $r \rightarrow \infty$  for  $i \neq 0$  and hence  $R(l) \rightarrow R_a(0) \delta_l$ .

### 12-9. The condition for periodicity is

$$b_{k+m} \oplus x_{k+m} = b_k \oplus x_k. \quad (12.113)$$

A sufficient condition for (12.113) to hold is that

$$m = ls = g_{h,k} \quad (12.114)$$

for some integers  $l$  and  $i$ . The smallest period  $m$  is by definition the LCM of  $s$  and  $r$ .

- 12-10. The shift-register output obeys difference equation  $x_k = x_{k-2}$ . It is easily shown that with either initial state (0,1) or (1,0) the output alternates between 0 and 1 (period two), and with initial state (1,1) the output is always one (period one).

- 12-11. We can consider that we are observing the first  $i$  bits out of  $n$  bits in the sequence. Given a particular  $i$ -bit sequence which is not all zeros, the remaining  $n-i$  bits will assume all of the  $2^{n-i}$  possible values. There are  $2^n - 1$  possible  $n$ -bit sequences, and therefore the relative frequency of seeing this particular  $i$ -bit sequence is the ratio of these two numbers. Similarly, if the  $i$ -bit sequence is all-zeros, then the remaining  $n-i$  bits cannot be all-zeros. Hence for this case there are only  $(2^{n-i} - 1)$  possible  $n-i$  bit sequences.



- 12-12. The maximal-length binary sequence  $x_k$  will have, in a period of  $r$  bits,  $\frac{r+1}{2}$  "ones" and  $\frac{r-1}{2}$  "zeros". Hence, the average of  $s_k$  over one period is

$$\mu_s = \frac{1}{r} \left[ \frac{r+1}{2} - \frac{r-1}{2} \right] = \frac{1}{r}. \quad (12.115)$$

- 12-13. In the autocorrelation definition of (12.86), the terms for which  $x_k = x_{k+l}$  contribute +1 while the remaining terms contribute -1. Hence the sum is

$$\frac{1}{r} \left( \frac{r-1}{2} - \frac{r+1}{2} \right) = -\frac{1}{r} \quad (12.116)$$

for  $1 \leq l \leq r-1$ . The answer is straightforward when  $l = 0$ .

- 12-14.

- (a) In evaluating the DFT, let  $n = k+l$ , so that the DFT of  $s_{k+l}$  becomes

$$e^{j \frac{2\pi m l}{r}} \sum_{n=l}^{l+r-1} s_n e^{-j \frac{2\pi m n}{r}}. \quad (12.117)$$

The sum can be split into two parts

$$\sum_{n=l}^{l+r-1} = \sum_{n=l}^{r-1} + \sum_{n=r}^{l+r-1} \quad (12.118)$$

where the second summation is

$$\begin{aligned} \sum_{n=r}^{l+r-1} s_n e^{-j \frac{2\pi m n}{r}} &= \sum_{k=0}^{l-1} s_{k+r} e^{-j \frac{2\pi m (k+r)}{r}} \\ &= \sum_{k=0}^{l-1} s_k e^{-j \frac{2\pi m k}{r}}. \end{aligned} \quad (12.119)$$

The substitution  $k = n-r$  was used, as well as the periodicity in  $r$  of all terms in the summation.

- (b) This is straightforward given the results of a., since

$$\begin{aligned} \sum_{l=0}^{r-1} \left( \frac{1}{r} \sum_{k=0}^{r-1} s_k s_{k+l} \right) e^{-j \frac{2\pi m l}{r}} &= \frac{1}{r} \sum_{k=0}^{r-1} s_k \sum_{l=0}^{r-1} s_{k+l} e^{-j \frac{2\pi m l}{r}} \\ &= e^{j \frac{2\pi m k}{r}} \sum_{l=0}^{r-1} s_l e^{-j \frac{2\pi m l}{r}}. \end{aligned} \quad (12.120)$$

- (c) Substituting from (12.89),

$$\begin{aligned} \sum_{l=0}^{r-1} R_s(l) e^{-j \frac{2\pi m l}{r}} &= 1 + \sum_{l=1}^{r-1} \left( -\frac{1}{r} \right) e^{-j \frac{2\pi m l}{r}} \\ &= 1 + \frac{1}{r} - \frac{1}{r} \sum_{l=0}^{r-1} e^{-j \frac{2\pi m l}{r}} \\ &= 1 + \frac{1}{r} \end{aligned} \quad (12.121)$$

when  $1 \leq m \leq r-1$ . When  $m = 0$  the value can be evaluated directly as  $\frac{1}{r}$ . We then multi-

ply by  $r$  to get  $|S_m|^2$ .

- 13-1. The square of the Euclidean distance is the square of the distance in one component  $(2a)^2$ , times the number of components that differ, which is the Hamming distance. The result follows immediately.

13-2.

- (a) Since between 1 and 4 bits are wrong when a block error occurs,

$$\frac{1}{4} \Pr[\text{block error}] \leq \Pr[\text{bit error}] \leq \Pr[\text{block error}] . \quad (13.110)$$

- (b) Combining the result of part (a) with (13.30), to get a probability of bit error of  $10^{-5}$  we need the BSC channel for the coded system to have a parameter  $p$  satisfying

$$0.0014 \geq p \geq 0.0007 . \quad (13.111)$$

Assuming binary antipodal signaling with alphabet  $\pm a$ , this means

$$0.0014 \geq Q\left[\frac{a}{\sigma_c}\right] \geq 0.0007 . \quad (13.112)$$

By successive approximation (using tables or an approximation for  $Q(\cdot)$  from Figure 3-1), we find

$$3.0 \leq \frac{a}{\sigma_c} \leq 3.2 . \quad (13.113)$$

Using (13.31) this means that for the coded system to achieve a bit error probability of  $10^{-5}$  with an alphabet  $\pm a$ , we require

$$3.97\sigma_u \leq a \leq 4.23\sigma_u . \quad (13.114)$$

The uncoded system achieves a bit error probability of  $10^{-5}$  when

$$a = 4.27\sigma_u \quad (13.115)$$

which implies the power advantage is

$$0.63\text{dB} \geq \text{power advantage} \geq 0.00\text{dB} . \quad (13.116)$$

- (c) Repeating for a probability of error of  $10^{-7}$ , we find that the coded system requires

$$0.00014 \geq p \geq 0.00007 , \quad (13.117)$$

$$0.00014 \geq Q\left[\frac{a}{\sigma_c}\right] \geq 0.00007 , \quad (13.118)$$

$$3.64 \leq \frac{a}{\sigma_c} \leq 3.82 , \quad (13.119)$$

$$4.82\sigma_u \leq a \leq 5.05\sigma_u . \quad (13.120)$$

The uncoded system achieves a bit error probability of  $10^{-7}$  when

$$a = 5.20\sigma_u \quad (13.121)$$

which implies the power advantage is

$$0.66\text{dB} \geq \text{power advantage} \geq 0.25\text{dB} . \quad (13.122)$$

- 13-3. We need to find the weight of the minimum weight codeword. This is easy to do if we observe that  $\mathbf{cH}' = \mathbf{0}$  is a linear combination of  $w_H(\mathbf{c})$  columns of  $\mathbf{H}$ . The minimum number of columns of  $\mathbf{H}$  that add to zero is  $d_{H, \min}$ , therefore. For Hamming codes, no two columns add to zero (because they would have to be identical). However, it is easy to find three columns that add to zero for any Hamming code. For example the columns

$$\begin{bmatrix} 0 \\ \dots \\ 0 \\ 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (13.123)$$

- 13-4. Verify that each of the cyclic shifts can be formed by linear combinations of rows of the generator matrix.
- 13-5. Do this by induction. The first  $m$  bits are source bits. The  $(m+1)^{\text{th}}$  bit is a modulo-two sum of a subset of the first  $m$  bits, and hence is a parity-check bit. The  $(m+2)^{\text{th}}$  bit is a modulo-two sum of a subset of the second through  $m^{\text{th}}$  source bits and the  $(m+1)^{\text{th}}$  bit, which is itself a modulo-two sum of the source bits, and hence is also a parity-check bit. Etc.
- 13-6. Define the state of the shift register to be the four bits in the shift register at any time. As long as the initial state is not zero, in the process of generating the codeword all  $2^m - 1$  nonzero  $m$  bit patterns will occur in the shift register. Hence a codeword consists of the rightmost bit of all of these  $m$  bit patterns, where the exact order is determined by the initial state. For all  $m$ , exactly  $2^{m-1}$  of these rightmost bits are ones, so the Hamming weight of the code must be  $2^{m-1}$ . This implies that the  $(2^m - 1, m)$  code with hard decoder can correct  $2^{m-2} - 1$  errors.
- 13-7. Note from Figure 13-8b that

$$C_k^{(3)} = C_k^{(1)} \oplus C_{k-1}^{(1)} \oplus C_{k-1}^{(2)} \quad (13.124)$$

or

$$C^{(3)}(D) = (1 \oplus D)C^{(1)}(D) \oplus DC^{(2)}(D). \quad (13.125)$$

Hence,

$$(1 \oplus D)C^{(1)}(D) \oplus DC^{(2)}(D) \oplus C^{(3)}(D) = 0 \quad (13.126)$$

which is what we wanted to verify.

- 13-8. From the distributive law,

$$r_1 * r_2 = r_1 * (r_2 \oplus 0) = r_1 * r_2 \oplus r_1 * 0, \quad (13.127)$$

so  $r_1 * 0 = 0$ , the additive identity.

- 13-9. All properties are easy to verify. The multiplicative inverses are:

element	inverse
0	none
1	1
2	3
3	2
4	4

- 13-10. There is no multiplicative inverse for 2 under modulo-four multiplication. Trying all possible values,  $2*0=0$ ,  $2*1=2$ ,  $2*2=0$ ,  $2*3=2$ , so there is no element in the field that we can multiply by 2 to get 1. A suitable multiplication table is

*	0	1	2
0	0	0	0
1	0	1	2
2	0	2	3
3	0	3	1

Note that there is a 1 in every non-zero row and column, so the multiplicative inverse exists for all elements in the field.

- 13-11. For any  $c \in C$ ,  $cH^T = 0$ , hence if  $h$  is a row of  $H$ , then  $ch^T = 0$ , and the vectors are orthogonal. As a consequence, if  $h$  is a row of  $H$  then  $hG^T = 0$ . Furthermore, if  $c_D$  is any linear combination of the rows of the rows of  $H$  (i.e. a codeword generated by  $H$ ), then  $c_D G^T = 0$ , so  $G$  is a parity check matrix for the code generated by  $H$ .
- 13-12. Addition and multiplication over  $F_D$  are ordinary polynomial addition and multiplication except that arithmetic on the coefficients is all modulo two. All the properties are easy to verify.

14-1.

- (a) The volume  $V(U^K)$  is given by the integral

$$V(U^K) = \int_U \int_U \cdots \int_U d\mathbf{x}_1 d\mathbf{x}_2 \cdots d\mathbf{x}_K, \quad (14.74)$$

and the integral can be written as the product of  $K$  integrals, each equal to  $V(U)$ .

(b)

$$P(U^K) = E \|\mathbf{X}\|^2 = \sum_{i=1}^K \|\mathbf{X}_i\|^2 = \sum_{i=1}^K E \|\mathbf{X}_i\|^2 = K \cdot P(U). \quad (14.75)$$

- 14-2. The average squared power of the 16-point constellation is 10 and of the 32-point constellation is 20, which is a 3 dB difference.
- 14-3. The 256 point constellation can be thought of as two successive points from the 16 point constellation and hence the average squared power is double that of the 16 point QAM constellation. From the previous exercise, the 16 point constellation has average squared power 10. Now we get the average squared power of the 512 point constellation. Note that the squared power of all symbols of the form (14.64) is  $25 + 1 + 1 + 1 = 28$ , and the squared power of all symbols of the form (14.65) is  $25 + 9 + 1 + 1 = 36$ , so the average squared power of the constellation is

$$\frac{1}{512} (256 \times 20 + 64 \times 28 + 192 \times 36) = 27. \quad (14.76)$$

- 14-4. Substituting (14.69), (14.19), and (14.20) into (14.70) we get

$$\gamma = \frac{NV^{2N}(S)d_{\min}^2(C)}{4V^{2N}(\Lambda)P(S)2^{\rho(C)}}. \quad (14.77)$$

Recall that the spectral efficiency  $\nu$  is the number of information bits communicated per two dimensions. Thus the constellation size is

$$2^{\nu + \rho(C)} = \frac{V^{2N}(S)}{V^{2N}(\Lambda)} \quad (14.78)$$

analogous to (14.12), so

$$\gamma = \frac{N 2^v d_{\min}^2(C)}{4P(S)}. \quad (14.79)$$

From (14.46) we can write this as

$$\gamma = \frac{2^v d_{\min}^2(C)}{2P}. \quad (14.80)$$

Recognizing that for large constellations  $2^v \approx 2^v - 1$ , (14.10) follows.

15-1. Assume the PLL is phase locked,

$$\phi(t) = \theta(t) + \phi = \omega_0 t + \theta + \phi. \quad (15.78)$$

Then from (15.5)

$$c(t) = \frac{d\phi(t)}{dt} = \omega_0. \quad (15.79)$$

From (15.7) and (15.3)

$$\varepsilon(t) = W(-\phi). \quad (15.80)$$

This is a constant (d.c.) so

$$c(t) = L(0)\varepsilon(t) = L(0)W(-\phi). \quad (15.81)$$

Comparing (15.81) with (15.79) we see that

$$\omega_0 = L(0)W(-\phi). \quad (15.82)$$

From Figure 15-3 we see that  $|W(-\phi)| \leq \pi$  so

$$|\omega_0| \leq \pi |L(0)|. \quad (15.83)$$

15-2.

$$\frac{\Phi(s)}{\Theta(s)} = \frac{N(s)/D(s)}{N(s)/D(s) + s} = \frac{N(s)}{N(s) + sD(s)} \quad (15.84)$$

from which the result follows.

15-3. By contradiction. Assume phase lock,

$$\phi(t) = \theta(t) = \omega_0 t + K \quad (15.85)$$

so

$$\frac{d\phi(t)}{dt} = \omega_0 \quad (15.86)$$

$$c(t) = \frac{d\phi(t)}{dt} = \omega_0. \quad (15.87)$$

This is a d.c. signal so

$$\varepsilon(t) = W(\phi(t) - \theta(t)) = \frac{1}{L(0)} c(t) = \frac{\omega_0}{L(0)} \neq 0 \quad (15.88)$$

so

$$\phi(t) \neq \theta(t). \quad (15.89)$$

15-4. Assume phase lock,

$$\phi_k = \theta_k + \phi = \omega_0 kT + \theta + \phi. \quad (15.90)$$

Consequently, from (15.37)

$$c_k = \phi_{k+1} - \phi_k = \omega_0 T. \quad (15.91)$$

The phase error is

$$\epsilon_k = W(\phi_k - \theta_k) = W(-\phi), \quad (15.92)$$

a d.c. signal, so

$$c_k = L(1)\epsilon_k = L(1)W(-\phi). \quad (15.93)$$

Combining (15.93) with (15.91) we get

$$\omega_0 = \frac{1}{T} L(1)W(-\phi) \quad (15.94)$$

so since  $|W(\cdot)| \leq \pi$ ,

$$|\omega_0| \leq \frac{\pi}{T} |L(1)|. \quad (15.95)$$

15-5. By inspection,

$$C(z) = L(z)(\Theta(z) - \Phi(z)) \quad (15.96)$$

and

$$\phi_{k+1} = c_k + \phi_k, \quad (15.97)$$

so

$$(z+1)\Phi(z) = C(z) = L(z)(\Theta(z) - \Phi(z)) \quad (15.98)$$

which easily reduces to the the desired result.

16-1. Multiply both sides of (16.7) by  $A_k^*$  and look at the imaginary part of both sides to get

$$\text{Im}\{q_k A_k^*\} = \sin(\epsilon_k) |A_k|^2. \quad (16.28)$$

The result follows easily.

16-2. Multiply both sides of (16.9) by  $A_k^*$  and examine the imaginary part to get

$$\epsilon_k = \sin^{-1} \left[ \frac{\text{Im}\{q_k A_k^*\}}{c_k |A_k|^2} \right]. \quad (16.29)$$

Then use the fact that  $|q_k| = c_k |A_k|$  to get the result.

16-3. In the decoder, after the initial conditions are cleared, the output of the  $z^{-1}$  boxes is exactly the same as the output of the  $z^{-1}$  boxes in the transmitter, regardless of  $M$ . Hence the subtractor removes what the adder inserted.

17-1.  $x(t)$  is periodic with period  $T$ , and so can be written as a Fourier series. From the results in appendix 15-A, the Fourier series coefficients are

$$X_m = \frac{E[A_k]}{T} P(j2\pi m/T), \quad (17.57)$$

which are scaled samples of the Fourier transform of the pulse  $P(j\omega)$ . For  $X(j\omega)$  to have a component at  $\omega = \pm 2\pi/T$ , it is necessary that  $X_1$  and  $X_{-1}$  be nonzero, which will only occur if  $P(j\omega)$  is nonzero at  $\omega = \pm 2\pi/T$ .

17-2.

- (a) The results of appendix 15-A apply directly, where

$$g(t) = |p(t)|^2, \quad (17.58)$$

and using the fact that multiplication in the time domain is equivalent to convolution in the frequency domain, and the fact that the Fourier transform of  $p^*(t)$  is  $P^*(-j\omega)$ , the result follows.

- (b) Writing

$$Z_{-n} = \frac{1}{2\pi} \int_{-\infty}^{\infty} P(j\omega) P^*(j(\omega + n\frac{2\pi}{T})) d\omega, \quad (17.59)$$

and changing variables by letting  $u = \omega + n\frac{2\pi}{T}$ , (17.59) becomes  $Z_n^*$ .

17-3. For  $m \neq n$ , by independence

$$E[A_m A_n] = E[A_m] E[A_n] = 0 \quad (17.60)$$

by the zero-mean assumption. For  $m = n$ , let  $A_m = C + jD$  then

$$A_m^2 = C^2 - D^2 + 2jCD \quad (17.61)$$

which has mean value zero since by assumption  $C$  and  $D$  have equal variance, and since the real and imaginary parts are independent they are uncorrelated.

17-4. Dropping the dependence on  $k$ , write  $E(\tau)$  in terms of its real and imaginary parts

$$E(\tau) = E_R(\tau) + jE_I(\tau) \quad (17.62)$$

and

$$|E(\tau)|^2 = E_R^2(\tau) + E_I^2(\tau) \quad (17.63)$$

so

$$\begin{aligned} \frac{\partial}{\partial \tau} |E(\tau)|^2 &= 2E_R(\tau) \frac{\partial E_R(\tau)}{\partial \tau} + 2E_I(\tau) \frac{\partial E_I(\tau)}{\partial \tau} \\ &= 2\text{Re} \left[ E^*(\tau) \frac{\partial E(\tau)}{\partial \tau} \right]. \end{aligned} \quad (17.64)$$

17-5.

- (a) It easy to show that
- $E[Q_k(\tau_k)]$
- is independent of
- $k$
- . To show that
- $R_{QQ}(k, i)$
- depends only on
- $k - i$
- we use the assumptions about
- $N_k$
- to write

$$R_{QQ}(k, i) = E \left[ \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} A_m A_n p((k-m)T + \tau_k) p((i-n)T + \tau_k) + N_k N_i \right]. \quad (17.65)$$

Using two variable changes,  $r = k - m$  and  $q = i - n$  we get

$$R_{QQ}(k, i) = \sum_{r=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} E[A_{k-r} A_{i-q}] p(rT + \tau_k) p(qT + \tau_k) + E[N_k N_i], \quad (17.66)$$

where we have also exchanged expectations with summations. This can be written

$$R_{QQ}(k, i) = \sum_{r=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} R_A(k - i - r + q) p(rT + \tau_k) p(qT + \tau_k) + R_N(k - i) \quad (17.67)$$

which depends only on  $k - i$ .

- (b) With  $Q_k(\tau_k)$  real valued, wide sense stationarity implies that

$$E[Q_k(\tau_k)Q_{k+1}(\tau_k)] = R_Q(1) \quad (17.68)$$

and

$$E[Q_k(\tau_k)Q_{k-1}(\tau_k)] = R_Q(-1). \quad (17.69)$$

From the symmetry of the autocorrelation function these are equal.

- 17-6. Plugging (17.32) into (17.31) we get that the timing function is

$$\begin{aligned} f(\tau_k) = & \operatorname{Re} \{ E[\hat{A}_k \sum_{m=-\infty}^{\infty} A_m p((k-m+1)T + \tau_k) + \hat{A}_k N_k] \\ & - E[\hat{A}_k \sum_{m=-\infty}^{\infty} A_m p((k-m-1)T + \tau_k) + \hat{A}_k N_k] \} \end{aligned} \quad (17.70)$$

Using the assumptions, this simplifies to get the result.

- 17-7. Exchange the integral and summation and change variables inside the integral, replacing  $t$  with  $t = \tau + kT$ , getting

$$\begin{aligned} X_m &= \frac{1}{T} \sum_{k=-\infty}^{\infty} \int_{-T/2}^{T/2} g(t - kT) e^{-j2\pi m t / T} dt \\ &= \frac{1}{T} \sum_{k=-\infty}^{\infty} \int_{-T/2 - kT}^{T/2 - kT} g(\tau) e^{-j2\pi m (\tau + kT) / T} d\tau. \end{aligned}$$

This can be simplified by observing that

$$e^{-j2\pi m (\tau + kT) / T} = e^{-j2\pi m \tau / T}. \quad (17.71)$$

Observe that the summation is a sum of finite integrals with adjoining limits, which may be replaced with a single infinite integral, getting the desired result.

- 18-1. We can define a fixed frame structure with a number of time-slots and added bit framing. One node of the ring is defined as the master, and it inserts the framing bits onto the ring. Each of the other nodes can detect this frame, thus defining the same frame and set of time slots. Now suppose a circuit is desired from station A to station B, and that time-slot  $n$  is allocated to this circuit. Then station A can identify time-slot  $n$ , and insert its information into that time-slot. For every other time-slot and the framing bits, station A simply retransmits whatever bits are incoming. At station B, it knows the position of time-slot  $n$  because it also has the framing. It therefore extracts the bits on this time-slot. Every other time-slot it also retransmits incoming bits. Note that a particular time-slot can be *re-used*; that is can support two or more circuits as long as those circuits don't overlap one another on the ring topology.

- 18-2. Taking the derivative w.r.t.  $\rho_{\text{out}}$ , we get

$$\frac{\partial \rho_X}{\partial \rho_{\text{out}}} = e^{-2\rho_{\text{out}}} - 2\rho_{\text{out}} e^{-2\rho_{\text{out}}} = 0 \quad (18.16)$$

which leads to  $\rho_{\text{out}} = 1/2$ .

- 19-1.

$$E[A_k e^{j\omega_c kT} A_m^* e^{-j\omega_c mT}] = e^{j\omega_c (k-m)T} E[A_k A_m^*] \quad (19.79)$$

$$R_A(l) = e^{j\omega_c lT} R_A(l) \quad (19.80)$$



- 19-2. The error can be written in the form, for a fixed coefficient vector,

$$E_k = R_k - e^{j\omega_c(k + \frac{l}{R})T} \mathbf{c}' \mathbf{a}_k. \quad (19.81)$$

Multiplying both sides by  $e^{-j\omega_c(k + \frac{l}{R})T}$  we get

$$e^{-j\omega_c(k + \frac{l}{R})T} E_k = e^{-j\omega_c(k + \frac{l}{R})T} R_k - \mathbf{c}' \mathbf{a}_k. \quad (19.82)$$

We see that this is the same problem as the passband transversal filter with the following substitutions:

$$R_k \rightarrow e^{-j\omega_c(k + \frac{l}{R})T} R_k \quad (19.83)$$

$$\bar{A}_k \rightarrow A_k \quad (19.84)$$

$$E_k \rightarrow e^{-j\omega_c(k + \frac{l}{R})T} E_k. \quad (19.85)$$

Making these substitutions we immediately get (19.34) and (19.35).

- 19-3. For a fixed coefficient vector, the real-error is

$$\text{Re}\{E_k\} = \text{Re}\{\mathbf{c}' \mathbf{a}_k\} \quad (19.86)$$

and taking the gradient with respect to the real-part of  $\mathbf{c}$ ,  $\mathbf{c}_R$ ,

$$\nabla_{\mathbf{c}_R} (\text{Re}\{E_k\})^2 = -2 \text{Re}\{E_k\} \text{Re}\{\mathbf{a}_k\} \quad (19.87)$$

and similarly taking the gradient with respect to the imaginary-part,

$$\nabla_{\mathbf{c}_I} (\text{Re}\{E_k\})^2 = 2 \text{Re}\{E_k\} \text{Im}\{\mathbf{a}_k\} \quad (19.88)$$

and hence the gradient with respect to  $\mathbf{c}$  is

$$\nabla_{\mathbf{c}} (\text{Re}\{E_k\})^2 = -2 \text{Re}\{E_k\} \mathbf{a}_k^* \quad (19.89)$$

- 19-4. Starting with (19.32), and substituting for  $E_k$ , we get

$$\mathbf{c}_k = \Gamma_k \mathbf{c}_{k-1} + \beta R_k \bar{\mathbf{a}}_k^* \quad (19.90)$$

Substituting  $\mathbf{q}_k + \mathbf{c}_{\text{opt}}$  for  $\mathbf{c}_k$ ,

$$\mathbf{q}_k = \Gamma_k \mathbf{q}_{k-1} - \beta \bar{\mathbf{a}}_k^* \bar{\mathbf{a}}_k' \mathbf{c}_{\text{opt}} + \beta R_k \bar{\mathbf{a}}_k^* \quad (19.91)$$

from which (19.41) follows immediately.

- 19-5. We have

$$\begin{aligned} \frac{\partial |E_k|^2}{\partial \hat{\theta}} &= \frac{\partial (E_k^* E_k)}{\partial \hat{\theta}} \\ &= E_k \frac{\partial E_k^*}{\partial \hat{\theta}} + E_k^* \frac{\partial E_k}{\partial \hat{\theta}} \\ &= -j e^{j\hat{\theta}} \mathbf{c}_k' \bar{\mathbf{a}}_k E_k^* + j e^{-j\hat{\theta}} \mathbf{c}_k^{*'} \bar{\mathbf{a}}_k^* E_k \\ &= -2 \text{Im}\{e^{-j\hat{\theta}} E_k \mathbf{c}_k' \bar{\mathbf{a}}_k\}. \end{aligned} \quad (19.92)$$

19-6. We get

$$E_k = (e^{j\theta_k} - e^{j\delta_k}) \mathbf{c}_k' \tilde{\mathbf{a}}_k. \quad (19.93)$$

The rest is straightforward algebra.

19-7. A typical element of the matrix  $\tilde{\mathbf{a}}_k \tilde{\mathbf{a}}_k'$  is  $\tilde{A}_{k-m} \tilde{A}_{k-n}^*$ . If  $m \neq n$  then this expectation is

$$E[\tilde{A}_{k-m} \tilde{A}_{k-n}^*] = E[\tilde{A}_{k-m}] E[\tilde{A}_{k-n}^*] = 0 \quad (19.94)$$

because of the independence assumption. For  $m=n$  if we write  $\tilde{A}_{k-m} = c + jd$  then

$$E[\tilde{A}_k^2] = E[c^2 - d^2 + 2jcd] = 0 \quad (19.95)$$

since

$$E[c^2] = E[d^2] \quad (19.96)$$

and

$$E[cd] = E[c]E[d] = 0. \quad (19.97)$$

The result for the uncanceled error follows immediately by the same method.

19-8. Starting with the SG algorithm of (19.36), and substituting  $\mathbf{q}_k + \mathbf{c}_{\text{opt}}$  for  $\mathbf{c}_k$ , we get immediately

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \beta \text{Re}\{R_k - \tilde{\mathbf{a}}_k' \mathbf{c}_{\text{opt}}\} \tilde{\mathbf{a}}_k^* - \beta \text{Re}\{\tilde{\mathbf{a}}_k' \mathbf{q}_{k-1}\} \tilde{\mathbf{a}}_k^*. \quad (19.98)$$

The result follows immediately after rewriting (19.98) in the form

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \beta \text{Re}\{D_k\} \tilde{\mathbf{a}}_k^* - \frac{1}{2} \beta (\tilde{\mathbf{a}}_k' \mathbf{q}_{k-1} + \tilde{\mathbf{a}}_k^{*'} \mathbf{q}_k^*) \tilde{\mathbf{a}}_k^*. \quad (19.99)$$

19-9. By definition,

$$\Gamma_k \Lambda_k^* = \frac{1}{2} \beta \tilde{\mathbf{a}}_k \tilde{\mathbf{a}}_k' - \frac{1}{4} \beta^2 (\tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k') (\tilde{\mathbf{a}}_k' \tilde{\mathbf{a}}_k) \quad (19.100)$$

and the expectation of the first term is zero in view of exercise 19-7. The expectation of a typical  $m, n$  term in the second stochastic matrix in (19.100) is

$$E \left[ \left( \sum_{j=1}^{N-1} \tilde{A}_{k-j}^2 \right) \tilde{A}_{k-m}^* \tilde{A}_{k-n} \right] \quad (19.101)$$

By the assumed independence, all terms in the sum except the  $j = m$  and  $j = n$  terms can be ignored since we know that  $E[\tilde{A}_{k-j}^2] = 0$ . Now if  $m \neq n$ , (19.101) becomes

$$E[\tilde{A}_{k-m}^2 \tilde{A}_{k-m}^* \tilde{A}_{k-n} + \tilde{A}_{k-n}^3 \tilde{A}_{k-m}^*] \quad (19.102)$$

and both terms are zero because of independence and since  $E[\tilde{A}_{k-m}] = E[\tilde{A}_{k-n}] = 0$  because of the zero-mean assumption. Thus, all that remains is the  $m = n$  case. The expectation of (19.102) reduces to

$$2E[\tilde{A}_{k-m}^2 |\tilde{A}_{k-m}|^2] \quad (19.103)$$

and letting  $\tilde{A}_k = c + jd$  this term reduces to

$$E[(c^2 + d^2)(c^2 - d^2 + 2jcd)] = E[(c^4 - d^4) + 2j(c^3d + cd^3)] = 0 \quad (19.104)$$

because

$$E[c^4] = E[d^4] \quad (19.105)$$

$$E[c^3d] = E[c^3]E[d] = 0. \quad (19.106)$$

19-10. By direct calculation

$$E[\Lambda_k^* \Lambda_k] = \frac{1}{4} \beta^2 E[\|\tilde{\mathbf{a}}_k\|^2 \tilde{\mathbf{a}}_k^* \tilde{\mathbf{a}}_k] \quad (19.107)$$

A typical  $m, n$  element of the stochastic matrix is

$$E\left[\left(\sum_{j=0}^{N-1} |\tilde{A}_{k-j}|^2\right) \tilde{A}_{k-m} \tilde{A}_{k-n}^*\right] \quad (19.108)$$

When  $m \neq n$  the expectation of all terms is zero by independence and the zero-mean property. Focusing then on the diagonal terms, (19.108) becomes

$$E\left[\left(\sum_{j=0}^{N-1} |\tilde{A}_{k-j}|^2\right) |\tilde{A}_{k-m}|^2\right] \quad (19.109)$$

and this expectation reduces to

$$\eta_a + (N-1)\sigma_a^4 \quad (19.110)$$

19-11.

(a) The uncanceled error is of the form

$$D_k = \mathbf{h}' \tilde{\mathbf{a}}_k + n_k \quad (19.111)$$

where  $\tilde{\mathbf{a}}_k$  and  $\mathbf{h}$  are a vector of far-end data symbols and the channel response respectively. Then we have

$$E[D_k^2] = E[\mathbf{h}' \tilde{\mathbf{a}}_k \tilde{\mathbf{a}}_k' \mathbf{h} + 2 \mathbf{h}' \tilde{\mathbf{a}}_k n_k + n_k^2] = 0 \quad (19.112)$$

in view of exercise 19-7.

(b) By straightforward calculation,

$$\begin{aligned} E[(\text{Re}\{D_k\})^2] &= \frac{1}{4} E[(D_k + D_k^*)^2] = \frac{1}{4} (E[D_k^2] + E[(D_k^*)^2] + 2E[|D_k|^2]) \\ &= \frac{1}{2} E[|D_k|^2] \end{aligned} \quad (19.113)$$

# INDEX

---

	Accumulation of timing jitter .....	757
	Adaptation of an echo canceler .....	809
	Adaptation training signal .....	519
	Adaptive equalization .....	198,212,517
Passband	adaptive equalization .....	546
Baseband	adaptive equalizer .....	520
	Adaptive filter coefficient drift .....	545
	Adaptive phase detectors .....	718
Automatic gain control,	AGC .....	183
Nyquist sampling theorem and	aliasing .....	16
	Allpass transfer functions .....	25
Pure and slotted	ALOHA .....	785
Channel	alphabet .....	105
Data symbols and	alphabet .....	181,213,380
	Alternate mark inversion (AMI) .....	297,564
	AM-DSB .....	20
Alternate mark inversion	(AMI) .....	297,564
Erbium-doped fiber	amplifiers .....	141
	Amplitude and phase modulation, AM/PM .....	203
Pulse	Amplitude Modulation (PAM) .....	12,182
Quadrature	amplitude modulation (QAM) .....	202,218,219
	Amplitude-phase modulation (AM-PM) .....	203,219
	AM-SSB .....	20
Economics of	analog and digital communication .....	6
	Analytic passband filter .....	204

	Analytic signal .....	18
Discrete-time	analytic signals .....	273
Isotropic	antenna .....	143
	Antenna gain and efficiency .....	143
Causal and	anti-causal sequences .....	21,273
Causal and	anti-causal systems .....	22
Binary	antipodal signal constellation .....	214,324,559
Maximum	a-posteriori (MAP) detector .....	381,379
	Aithmetic and geometric means .....	444
Optical fiber material	attenuation .....	131
Radio system margin and rain	attenuation .....	148
Propagation,	attenuation, and phase constants .....	120
	Attenuation of voiceband channel .....	162
Complementary	autocorrelation function .....	312
	Autocorrelation function of a pulse .....	47
	Autocorrelation function of a random process .....	58
	Autocorrelation matrix .....	522
	Autocorrelation matrix .....	551
Mean and	autocorrelation of a random process .....	57
	Automatic gain control, AGC .....	183
	Available noise power .....	140
Photodetectors, PIN diode and	avalanche photodiode .....	136
Bipolar six-zero substitution	(B6ZS) .....	605
Excess	bandwidth .....	188,191
Water-pouring	bandwidth .....	491
	Baseband adaptive equalizer .....	520
	Baseband PAM .....	191
Baseline wander in	baseband PAM systems .....	557
Equivalent	baseband pulse .....	208
	Baseband transversal filter .....	807
	Baseline wander .....	558
Symbol rate,	baud rate, bit rate .....	181
	Baud-rate timing recovery .....	754
	Bayesian detection .....	52,379,381
Bose-Chaudhuri-Hocquenghem codes	(BCH) codes .....	624
	BIBO stability .....	23
	Bimode line code .....	571
	Binary antipodal signal constellation .....	214,324,559
	Binary phase-shift keying (BPSK) .....	214,219
Differential	binary PSK (DBPSK) .....	221
	Binary symmetric channel (BSC) .....	102,383,412,634
	Biphase or Manchester pulse shape .....	559
High-density	bipolar (HDBk) line codes .....	605
Alternate mark inversion (AMI) or	bipolar line code .....	564
	Bipolar six-zero substitution (B6ZS) .....	605
	Bipolar violation .....	604
	Birth and death process .....	76
Symbol rate, baud rate,	bit rate .....	181
	Bit stuffing code .....	573,778
	Bits per second, b/s .....	185
Linear	block codes .....	639
Probability of	block error, block error rate .....	328
Efficiency of a	block line code .....	569
	Block line codes .....	568
	Bose-Chaudhuri-Hocquenghem codes (BCH) codes .....	624

Binary phase-shift keying,	BPSK .....	214,219
Trellis diagram nodes,	Branch and path metric for the Viterbi .....	414
	branches, and paths .....	413
	Bridged taps .....	172
	Broadband channel model .....	149
	Broadcast medium .....	768
Binary symmetric channel	(BSC) .....	102,383,412,634
Probability of error for	BSC .....	389
Link power	budget .....	144
First-in first-out (FIFO)	buffer .....	781
Queueing,	buffers, waiting positions, and servers .....	80
	Burst errors .....	624
	Bus, ring, and tree topologies .....	768
	Bytes and octets .....	771
Twisted pair and coaxial	cable .....	119
Primary and secondary parameters of	cable .....	124
	Campbell's theorem .....	84
	ISI canceler .....	511
	Passband echo canceler .....	804
	Nyquist echo canceler .....	808
	Adaptation of an echo canceler .....	809
	Echo suppressors and cancelers .....	166
	Interleaved echo cancelers .....	802
	Echo cancellation .....	167,766,797
	Far-end echo cancellation .....	815
	Channel capacity .....	103
Water-pouring spectrum for channel	capacity .....	491
Channel	capacity .....	99,669
	Capacity of additive Gaussian noise channel .....	108,348,489
	Capacity of vector Gaussian noise channel .....	108
	Capture of PLLs .....	709
	Carrier frequency .....	200
	Carrier recovery .....	211,725
	carrier recovery .....	726
Decision-directed	carrier recovery .....	735
False lock in	Carrier-sense multiple access (CSMA) .....	787
	Catastrophic codes .....	423,679
	Causal and anti-causal sequences and systems .....	21
	CCITT V.21 standard .....	240
	CCITT V.29 standard .....	373
Code-division multiple access	(CDMA) .....	261,789
Power control in	CDMA .....	791
	Cellular mobile radio .....	142,793
IS-54 standard for digital	cellular radio .....	221
Digital	cellular telephone .....	792
	Central limit theorem .....	55,250
	Markov chains and shift-register process .....	410
Voice-frequency (VF)	channel .....	116
	Channel alphabet .....	105
	channel (BSC) .....	102,383,412,634
	channel capacity .....	257
	channel capacity .....	491
	Channel capacity .....	99,103,669
	channel coding .....	98
	Channel capacity and channel coding theorem .....	98

Broadband	channel model	149
Narrowband	channel model	149
Memoryless discrete-time	channels	102
Physical media and	channels	115
Composite	channels	116
Delay spread for multipath	channels	149
Telephone	channels	160
Satellite	channels	240
	Characteristic impedance	120
	Characteristic or moment generating function	51
Parity	check matrix of convolutional codes	627
	Chernoff bound	51
	Chip waveform in spread spectrum	339
Optical fiber	chromatic or material dispersion	131
Exclusive-or	circuit	564
Parity	circuit	69
	Circuit and packet switching	5
Digital	circuit switch	773
	Circular convolution	258
	Circularly symmetric Gaussian process	313
Optical fiber core,	cladding, and sheath	129
PLL	clicks	702
	C-notched and C-message noise	163
Twisted pair and	coaxial cable	119
Gray	code	328
Twinned binary line	code	561
Alternate mark inversion (AMI) or bipolar line	code	564
Efficiency of a block line	code	569
Modes of a sequence-state line	code	569
Bimode line	code	571
Zero-disparity line	code	571
Bit stuffing	code	573
Convolutional	code	626
Lattice	code	652
Spreading	code in spread spectrum	339, 374
Pulse	code modulation (PCM)	2
	Code-division multiple access (CDMA)	261, 766, 789
Convolutional	coders	412
Sequence-state line	coders	566
Systematic convolutional	coders	626
Line	codes	555
Block line	codes	568
Signal-space	codes	610
Parity-check	codes	613
Hamming	codes	614
Cyclic	codes	624
Maximal-length shift register	codes	624
Reed-Solomon	codes	624
Linear block	codes	639
Repetition	codes	645
Signal-space	codes	650
Catastrophic	codes	679
Lattice and coset	codes	684
Bose-Chaudhuri-Hocquenghem	codes (BCH) codes	624
Tomlinson-Harashima	coding	460

Correlative coding	582
Channel or error-control coding	609
Trellis coding	668
Bit stuffing coding	778
Variable-rate coding	778
Source and channel coding	98
Coding gain	612
Shaping and coding gain	652
Source coding theorem	101
Channel capacity and channel coding theorem	98
Error coefficient	320
Correlation coefficient	55
Adaptive filter coefficient drift	545
Coefficient leakage in the FSE	545
Tap weights or coefficients	487
Coherent and incoherent receivers	247
Coherent demodulation	725
Coherent optical fiber reception	365,789
Coherent (synchrony) and differential	222
Collisions	767,770
Communications link	767
Complementary autocorrelation function	312
Complementary error function	53
Complementary probability distribution function	53
Complex exponential	12
Real and complex LTI systems	13
Complex-valued Gaussian process	312
Complex-valued signals	12
Composite channels	116
Data or computer network	4
Computer communication	4
Conditional entropy	99,102
Conditional probability	52
Connection in packet switching	779
Circuits or connections	770
Constant envelope	240,240
Propagation, attenuation, and phase constants	120
Definition of signal constellation	213
Binary antipodal signal constellation	214,324
Multidimensional signal constellation	651
Multidimensional signal constellation	220,654
Cross constellations	218
Hexagonal constellations	219
Rectangular constellations	219
Contention	767
Continuous approximation	463,575,659
Continuous-phase FSK (CPFSK)	243
Continuous-phase modulation (CPM)	243,589
Continuous-time matched filter	396
Mixed discrete and continuous-time PLLs	715
Convergence of MSEG algorithm	526
Convergence of SG algorithm	535
Region of convergence (ROC)	22
Circular convolution	258
Convolutional coders	412,626



Generator matrix of	convolutional coders .....	627
Viterbi algorithm applied to	convolutional coders .....	631
Optical fiber	core, cladding, and sheath .....	129
	Correlation coefficient .....	55
	Correlation receiver .....	225,252,289
Discrete-time	correlation receiver .....	257
	Correlation receiver for orthogonal multipulse .....	232
	Correlative coding .....	582
Lattice and	coset codes .....	684
	Coset representative .....	685
Square-root raised	cosine pulses .....	228
	Counting process .....	75
	Covariance matrix .....	56
Continuous-phase FSK	(CPFSK) .....	243
Continuous-phase modulation	(CPM) .....	243,589
Full response	CPM .....	590
Partial-response	CPM .....	590
Phase diagrams for	CPM .....	590
	Cross constellations .....	218
	Cross-correlation .....	60,391
	Cross-correlation function of a random process .....	61
	Cross-correlation of random variables .....	51
	Cross-spectral density of a random process .....	61
	Crosstalk .....	162
Near- and far-end	crosstalk (NEXT and FEXT) .....	127
Carrier-sense multiple access	(CSMA) .....	787
	Cycle-and-add property of maximal-length .....	601
	Cyclic codes .....	624
	Cyclic extension .....	259
	Cyclostationary random process .....	65,86,741
Voiceband	data modems .....	186,769
	Data or computer network .....	4
Rotated	data symbols .....	547,805
	Data symbols and alphabet .....	181,380
Multitone	data transmission .....	256
Full-duplex	data transmission .....	789
Differential binary PSK	(DBPSK) .....	221
Birth and	death process .....	76
Quantizer, slicer,	decision .....	184
	Decision feedback equalizer (DFE) .....	541
Slicer	decision regions .....	215,288
	Decision regions .....	387
	Decision-directed carrier recovery .....	726
	Decision-directed timing recovery .....	750
	Decision-direction .....	212
Parallel	decision-feedback equalization .....	693
	Decision-feedback equalizer (DFE) .....	198,211,473,689
Mean-square error	decision-feedback equalizer (DFE-MSE) .....	477
Zero-forcing	decision-feedback equalizer (DFE-ZF) .....	453,475,691
Hard and soft	decoding .....	610,650
Modal	decomposition .....	527
	Deductive and inductive timing recovery .....	738
Propagation and processing	delay .....	3
Queueing	delay .....	780
Group	delay and dispersion .....	126

Queueing	delay and waiting time .....	82
Envelope	delay distortion .....	162
	Delay spread for multipath channels .....	149
Dirac	delta or impulse function .....	11,49
	Demand assignment in TDMA .....	775
	Demodulation .....	200
Coherent	demodulation .....	725
One's	density .....	605
Probability	density function .....	49
Power spectral	density of a random process .....	58
Cross-spectral	density of a random process .....	61
Generalized	derivative .....	49
Discrete-time approximation to	derivative .....	760
Coherent (synchrony) and differential	detection .....	222
Estimation and	detection .....	378
Maximum likelihood (ML)	detection .....	380
Direct	detection in optical fiber systems .....	138
ML sequence	detection (MLSD) .....	409
	Detection of a vector signal .....	385
Reduced-state sequence	detection (RSSD) .....	692
Viterbi	detector .....	198
Maximum a-posteriori (MAP)	detector .....	381
MAP sequence	detector .....	430
ML sequence	detector .....	431
Phase	detector .....	701
MAP	detector for vectors .....	388
Maximum-likelihood sequence	detector (MLSD) .....	406,409,442,448,480
Frequency	detectors .....	702
Sampling phase	detectors .....	715
Peak	deviation in FSK .....	242
Distributed-feedback	(DFB) lasers .....	135
Decision-feedback equalizer	(DFE) .....	211,473,541,689
	DFE error propagation .....	456,511
	DFE postcursor equalizer .....	456
Mean-square error decision-feedback equalizer	(DFE-MSE) .....	477
Unbiased	DFE-MSE (DFE-MSE-U) .....	479
Unbiased DFE-MSE	(DFE-MSE-U) .....	479
Zero-forcing decision-feedback equalizer	(DFE-ZF) .....	453,475,691
	DFT, FFT, inverse FFT, IFFT .....	256
	Dicode partial response .....	579,583
	Differential binary PSK (DBPSK) .....	221
Coherent (synchrony) and	differential detection .....	222
	Differential encoding .....	220,564
Error propagation in	differential encoding .....	729
	Differential PSK (DPSK) .....	221
IS-54 standard for	digital cellular radio .....	221
	Digital cellular telephone .....	792
	Digital circuit switch .....	773
Definition of	digital communication .....	4
Economics of analog and	digital communication .....	6
	Digital communications network .....	765
	Digital magnetic recording .....	167
Integrated services	digital network (ISDN) .....	4,119
	Digital PLLs .....	715
	Digital radio .....	142

	Digital subscriber loop .....	119,789,797
Running	digital sum (RDS) .....	558,575
	Digital sum variation (DSV) .....	562
Light-emitting	diode and semiconductor laser .....	135
	Direct detection in optical fiber systems .....	138
	Direct-sequence spread spectrum .....	341,790
Mixed	discrete and continuous-time PLLs .....	715
	Discrete-time analytic signals .....	273
	Discrete-time approximation to derivative .....	760
Memoryless	discrete-time channels .....	102
	Discrete-time correlation receiver .....	257
	Discrete-time Fourier transform .....	14
	Discrete-time Hilbert transform .....	273
	Discrete-time matched filter .....	391
	Discrete-time PLL .....	709
Group delay and	dispersion .....	126
Optical fiber chromatic or material	dispersion .....	131
Optical fiber mode	dispersion .....	131
	Dispersion and intersymbol interference .....	465
Minimum	distance .....	215,656
Hamming	distance .....	387,639
Intermodulation	distortion .....	118
Aliasing	distortion .....	16
Slope	distortion .....	161
Envelope delay	distortion .....	162
Source coding and rate	distortion .....	98
	Distributed-feedback (DFB) lasers .....	135
Complementary probability	distribution function .....	53
Cumulative	distribution function of a random variable .....	49
Marginal	distributions .....	50
Joint	distributions of random variables .....	50
	Diversity .....	152
	Doppler shift .....	154
Differential PSK	(DPSK) .....	221
Digital sum variation	(DSV) .....	562
	DTFT .....	14
Modulo-two	D-transform .....	593
	Dual codes .....	640
	Duobinary partial response .....	579,584,740
Modified	duobinary partial response coding .....	580
Full and half	duplex .....	167
Echo cancellation	(EC) .....	167
Talker and listener	echo .....	166
Passband	echo canceler .....	804
Nyquist	echo canceler .....	808
Adaptation of an	echo canceler .....	809
Interleaved	echo cancelers .....	802
	Echo cancellation .....	167,766,797
Far-end	echo cancellation .....	815
	Echo suppressors and cancelers .....	166
Antenna gain and	efficiency .....	143
Spectral	efficiency .....	185,304,349
Quantum	efficiency and responsivity .....	136
Spectral	efficiency of PAM .....	223
	Eigenfunctions and eigenvalues .....	43

Matrix	eigenvalue spread .....	528,551
Matrix	eigenvalues .....	526,544
Spectral	emission masks .....	147
	Energy of a signal .....	13,44
PAM equalizer noise	enhancement .....	198
Noise	enhancement in equalization .....	210,323,450,320,442,468
	Entropy .....	98
Constant	envelope .....	240,589
	Envelope delay distortion .....	162
	Envelope derived timing .....	747
Noise enhancement in	equalization .....	320,442,468
Adaptive	equalization .....	517
Decision-direction	equalization .....	518
Passband adaptive	equalization .....	546
Parallel decision-feedback	equalization .....	693
Decision-directed	equalization for multicarrier modulation .....	259
Zero-forcing linear	equalization (LE-ZF) .....	469
	Equalization noise enhancement .....	210
Adaptive	equalizer .....	198
Decision-feedback	equalizer .....	198
Fractionally-spaced forward	equalizer .....	211
Precursor	equalizer .....	299,33;5
DFE precursor	equalizer .....	455
DFE postcursor	equalizer .....	456
Baseband adaptive	equalizer .....	520
Decision-feedback	equalizer (DFE) .....	211,473,541,689
Mean-square error decision-feedback	equalizer (DFE-MSE) .....	477
Zero-forcing decision-feedback	equalizer (DFE-ZF) .....	453,475,691
Fractionally spaced	equalizer (FSE) .....	482,484,544
Linear	equalizer (LE) .....	468,689
Mean-square error linear	equalizer (LE-MSE) .....	471
Zero-forcing linear	equalizer (LE-ZF) .....	451,469
PAM	equalizer noise enhancement .....	198
Adaptive	equalizers .....	212
Constrained-complexity	equalizers .....	519
Fractionally-spaced	equalizers .....	753
Asymptotic	equipartition theorem .....	100,110
	Erbium-doped fiber amplifiers .....	141
Probability of	error .....	3,382
Symbol	error .....	324
Union bound on the probability of	error .....	327
Probability of block	error, block error rate .....	328
	Error coefficient .....	320
	Error events in the ML sequence detector .....	418
	Error events vs. bit and symbol error .....	432
Probability of	error for BSC .....	389
Complementary	error function .....	53
Excess mean-square	error (MSE) .....	536
Sequence	error probability .....	418
Error events vs. bit and symbol	error probability .....	432
DFE	error propagation .....	456,460,511
	Error propagation in differential encoding .....	729
Probability of block error, block	error rate .....	328
Channel or	error-control coding .....	609
Burst	errors .....	624

	Estimation and detection .....	378
Minimum mean-square error (MSE)	estimator .....	521
	Ethernet .....	787
	Euclidean space .....	32
	Excess bandwidth .....	188,191
	Excess mean-square error (MSE) .....	536
	Exclusive-or circuit .....	564
	Exclusive-or phase detectors .....	716
Fundamental theorem of	expectation .....	50
	Expected or mean value .....	50
Complex	exponential .....	12
Cyclic	extension .....	259
Monic minimum-phase spectral	Eye diagram .....	191
Multipath and selective	factorization .....	31,62,458
Rayleigh	fading .....	148
Frequency selective multipath	fading .....	152
	fading .....	230
	False lock in carrier recovery .....	735
Near- and	far-end crosstalk (NEXT and FEXT) .....	127
	Far-end echo cancellation .....	815
	Fast FSK (FFSK) .....	244
Frequency-division multiplexing	(FDM) .....	117,167,797
Frequency-division multiple access	(FDMA) .....	260,787
Quantized	feedback .....	592
	Feedback shift registers .....	592
Fast FSK	(FFSK) .....	244
DFT, FFT, inverse	FFT, IFFT .....	256
Optical	fiber .....	128
Graded-index	fiber .....	130
Multimode and single mode optical	fiber .....	130
Receiver design for optical	fiber .....	261,360
Coherent optical	fiber .....	789
Erbium-doped	fiber amplifiers .....	141
Optical	fiber chromatic or material dispersion .....	131
Optical	fiber core, cladding, and sheath .....	129
Optical	fiber material attenuation .....	131
Optical	fiber mode dispersion .....	131
Quantum limit for	fiber optical transmission .....	361
Coherent optical	fiber reception .....	365
Homodyne and heterodyne optical	fiber reception .....	365
Direct detection in optical	fiber systems .....	138
Finite or Galois	fields .....	598,638
First-in first-out	(FIFO) buffer .....	781
Analytic passband	filter .....	204
Causal matched	filter .....	227
Hilbert transform	filter .....	272
Sampled matched	filter .....	295
Discrete-time matched	filter .....	391
Whitening	filter .....	392
Continuous-time matched	filter .....	396
Phase shift	filter .....	44
Transversal	filter .....	487
Finite transversal	filter .....	518
Lattice	filter .....	549
Whitening	filter .....	62

Proportional plus integral loop	filter	708
Baseband transversal	filter	807
Correlation and matched	filter receiver	234,289
Whitened matched	filter (WMF)	406,442,445
Shot noise or	filtered Poisson process	83
Spectral nulls using	filtering	574
Transversal	filters	520
	Final value theorem for Laplace transforms	708
	Final value theorem for Z transforms	712
	Finite impulse response (FIR)	25
	Finite-state machine (FSM)	409
Finite impulse response	(FIR)	25
	First-in first-out (FIFO) buffer	781
	Flow control	779
Signal	flow graphs	71
	Folded spectrum	228,290
Fractionally-spaced	forward equalizer	211
	Fourier transform	13
Discrete-time	Fourier transform	14
	Fractionally spaced equalizer (FSE)	482,484,544,753
TDM	frame and framing bits	771
Link	frame in packet switching	778
	Frame synchronization	594,771
TDM frame and	framing bits	771
Carrier	frequency	200
	Frequency detectors	702
	Frequency division multiplexing (FDM)	167,256
	Frequency offset	726
	Frequency response	14,45
	Frequency selective multipath fading	230
	Frequency separation in FSK	242
	Frequency shift keying (FSK)	238
	Frequency synthesizers	719
	Frequency-division multiple access (FDMA)	260,787
Time- and	frequency-division multiplexing	116,797
	Friis transmission equation	144
Fractionally spaced equalizer	(FSE)	482,484,544
Coefficient leakage in the	FSE	545
Frequency shift keying	(FSK)	238
Binary	FSK	239
Frequency separation in	FSK	242
Peak deviation in	FSK	242
Continuous-phase	FSK (CPFSK)	243
Fast	FSK (FFSK)	244
Finite-state machine	(FSM)	409
	Full and half duplex	167
	Full response CPM	590
	Full-duplex and half-duplex transmission	768
	Full-duplex modems	185
	Fundamental volume of a lattice	656
Shaping and coding	gain	612,652
Antenna	gain and efficiency	143
Automatic	gain control, AGC	183
Processing	gain in spread spectrum	343
Finite or	Galois fields	598,638

Vector space over	Galois fields .....	639
Capacity of an ideal	Gaussian channel .....	348
Capacity of continuous-time	Gaussian channel .....	489
QO function, tail of a	Gaussian distribution .....	53
Additive	Gaussian noise channel .....	104
Capacity of additive	Gaussian noise channel .....	108
Capacity of vector	Gaussian noise channel .....	108
	Gaussian or normal random variables .....	53
Circularly symmetric	Gaussian process .....	313
	Gaussian random process .....	57
Jointly	Gaussian random variables .....	55
	Gaussian white noise .....	61
	Gear-shift algorithm .....	540
	Generalized Nyquist criterion .....	235,266,304
Characteristic or moment	generating function .....	51
Moment	generating function of shot noise .....	91
	Generator matrix .....	613,627
	Generator polynomial .....	593,624
Arithmetic and	geometric means .....	31,444
	Geometric structure of signal space .....	34
	Graded-index fiber .....	130
Step size of	gradient algorithm .....	526
Stochastic	gradient algorithm in timing recovery .....	749
MSE	gradient algorithm (MSEG) .....	525
Stochastic	gradient (SG) algorithm .....	532,812
	Gray code .....	328
	Group delay and dispersion .....	126
Full and	half duplex .....	167,768
	Hamming codes .....	614
	Hamming weight and distance .....	387,639
	Hard and soft decoding .....	610,650
High-density bipolar	(HDBk) line codes .....	605
	HDLC .....	263,778
	Hermitian and Toeplitz matrices .....	522
Homodyne and	heterodyne optical fiber reception .....	365
	Hexagonal constellations .....	219
Inner product and	Hilbert space .....	36
Discrete-time	Hilbert transform .....	273
	Hilbert transform filter .....	272
	Hilbert transforms in the frequency domain .....	273
Lock or	hold-in range of a PLL .....	704
	Homodyne and heterodyne optical fiber reception .....	365
Roll-call vs.	hub polling .....	783
	Hybrid for two- to four-wire conversion .....	165
Capacity of an	ideal Gaussian channel .....	348
Independent and	identically distributed (i.i.d.) .....	55
DFT, FFT, inverse FFT,	IFFT .....	256
Infinite impulse response	(IIR) .....	25
Characteristic	impedance .....	120
Dirac delta or	impulse function .....	11,49
Thermal, quantization, and	impulse noise .....	162,257
Coherent and	incoherent receivers .....	240,247
Statistical	independence .....	50
	independent and identically distributed .....	55
Modulation	index .....	589

	Infinite impulse response (IIR) .....	25
Average mutual information .....	information .....	103
Mutual information .....	information .....	98
Shannon information theory .....	information theory .....	97
	Inner product and Hilbert space .....	36
Vector inner product and norm .....	inner product and norm .....	34
	Innovations process .....	62
	In-phase and quadrature component .....	202
	Integrated Services Digital Network (ISDN) .....	4,119
Jamming and interference .....	interference .....	343
Intersymbol interference (ISI) .....	interference (ISI) .....	131,213,188,442,465
	Intermodulation distortion .....	118
Total internal reflection .....	internal reflection .....	128
Definition of intersymbol interference (ISI) .....	intersymbol interference (ISI) .....	189
DFT, FFT, inverse FFT, IFFT .....	inverse FFT, IFFT .....	256
Alternate mark inversion (AMI) .....	inversion (AMI) .....	297
Alternate mark inversion (AMI) or bipolar line code .....	inversion (AMI) or bipolar line code .....	564
	Irreducible polynomials .....	598
	IS-54 standard for digital cellular radio .....	221
Integrated Services Digital Network (ISDN) .....	(ISDN) .....	4,119
	Isolated pulse .....	225
	Isotropic antenna .....	143
	Jamming and interference .....	230,343
	Jensen's inequality .....	112
Timing jitter .....	jitter .....	3,738
Phase jitter .....	jitter .....	726
Accumulation of timing jitter .....	jitter .....	757
Granular noise and zero-crossing jitter in magnetic recording .....	jitter in magnetic recording .....	171
	Karhunen-Loeve expansion .....	393
Local and metropolitan area networks (LAN and MAN) .....	(LAN and MAN) .....	4
	Landau-Pollak theorem .....	305
	Laplace transforms .....	708
Light-emitting diode and semiconductor laser .....	laser .....	135
Linewidth of a laser .....	laser .....	371
Distributed-feedback (DFB) lasers .....	lasers .....	135
Fundamental volume of a lattice .....	lattice .....	656
	Lattice and coset codes .....	652,655,684
	Lattice filter .....	549
	Lattice partition .....	685
Linear equalizer (LE) .....	(LE) .....	468,689
Coefficient leakage in the FSE .....	leakage in the FSE .....	545
	Leased and conditioned lines .....	167
	Least-square (LS) algorithms .....	549
	left-sided sequences .....	21
Right-sided and (LE-MSE) .....	(LE-MSE) .....	471
Mean-square error linear equalizer (LE-ZF) .....	(LE-ZF) .....	451,469
Zero-forcing linear equalizer .....	Light-emitting diode and semiconductor laser .....	135
	likelihood (ML) detection .....	380
Maximum line codes .....	line codes .....	555
Twinned binary line code .....	line code .....	561
Alternate mark inversion (AMI) or bipolar line code .....	line code .....	564
Efficiency of a block line code .....	line code .....	569
Bimode line code .....	line code .....	571
Zero-disparity line code .....	line code .....	571
Sequence-state line code .....	line code .....	566,569



Pseudoternary	line code	560
Interleaved	line code	562
Block	line code	568
High-density bipolar (HDBk)	line code	605
	Linear block codes	639
	Linear equalizer (LE)	468,689
Mean-square error	linear equalizer (LE-MSE)	471
Zero-forcing	linear equalizer (LE-ZF)	451,469
	Linear prediction	62,459,473
	Linear space or vector space	31
Subspace of a	linear space	37
	Linear time-invariant (LTI) systems	13
	Linearity of codes	637
Optical source	linewidth	131,371
	Link frame in packet switching	778
	Link power budget	144
	Link utilization	782
Talker and	listener echo	166
	LMS timing recovery	748
	LMS transversal filter algorithm	532
	Local and metropolitan area networks (LAN and	4
	Local-area networks	766
	Lock or hold-in range of a PLL	704
False	lock in carrier recovery	735
Quadrature phase	lock of sinusoidal phase detectors	714
Strictly and	loosely maximum-phase transfer functions	27
Least-square	(LS) algorithms	549
Linear time-invariant	(LTI) systems	13
Real and complex	LTI systems	13
Eigenfunction of an	LTI system	45
Frequency response of	LTI system	45
	Magnetic and optical storage	4
Digital	magnetic recording	167,262
A.c.-bias and saturation	magnetic recording	169
Granular noise and zero-crossing jitter in	magnetic recording	171
	Magnitude response	15
Local and metropolitan area networks (LAN and	MAN)	4
Biphase or	Manchester pulse shape	559
Maximum a-posteriori probability	(MAP)	379
Maximum a-posteriori	(MAP) detector	381
	MAP detector for vectors	388
	MAP sequence detector	430
	Mapping by set partitioning	688
Radio system	margin and rain attenuation	148
	Marginal distributions	50
	Marginal probability	52
Markov process and	Markov chain	68-74,79
Power spectrum of a	Markov chain	87
	Markov chains and shift-register process	410
Spectral emission	masks	147
	Matched filter	224-229,234
Causal	matched filter	227
Sampled	matched filter	295
Discrete-time correlation or	matched-filter receiver	255,391
Continuous-time	matched filter	396

	Matched filter bound on the SNR .....	226
Correlation and	matched filter receiver .....	225,289
Whitened	matched filter (WMF) .....	406,442,445
	Matched-filter bound .....	448
	Matrix spectral decomposition .....	527
Cycle-and-add property of	maximal-length sequences .....	601
	Maximal-length shift register (MLSR) .....	591,
	Maximal-length shift register codes .....	624
	Maximum a-posteriori probability (MAP) .....	379,381
	Maximum likelihood (ML) detection .....	379,380
	ML sequence detector (MLSD) .....	406,409-423,442,448,480
Strictly and loosely	maximum-phase transfer functions .....	27
Multicarrier modulation	(MCM) .....	693
Arithmetic and geometric	mean .....	31,444
	Mean and autocorrelation of a random process .....	57
	Mean and variance .....	50
Expected or	mean value .....	50
	Mean-square error (MSE) .....	467,469
	Mean-square error decision-feedback equalizer .....	477
	Mean-square error linear equalizer (LE-MSE) .....	471
Excess	mean-square error (MSE) .....	536
Minimum	mean-square error (MSE) estimator .....	521
Physical	media and channels .....	115
Broadcast	medium .....	768
	Medium topology .....	767
	Memoryless discrete-time channels .....	102
Survivor and	merged paths for the Viterbi algorithm .....	416
Branch and path	metric for the Viterbi algorithm .....	414
Local and	metropolitan area networks (LAN and MAN) .....	4
	Microwave radio .....	142,788
	Minimum distance .....	215,280,656
	Minimum mean-square error (MSE) estimator .....	521
Monic	minimum-phase spectral factorization .....	31,62,458
Strictly and loosely	minimum-phase transfer functions .....	27
	Minimum-shift keying (MSK) .....	244,589
Maximum likelihood	(ML) detection .....	380
	ML sequence detection (MLSD) .....	406,409,431,442,480
Error events in the	ML sequence detector .....	418
	ML timing recovery .....	748
Maximum-length shift register	(MLSR) .....	591
	MMSE timing recovery .....	748
Cellular	mobile radio .....	142,793
	Modal decomposition .....	527
	Modal matrix .....	526
Optical fiber	mode dispersion .....	131
Multimode and single	mode optical fiber .....	130
Definition of	modem .....	179
Voiceband data	modem .....	186,769
Full-duplex	modem .....	185
	Modes of a sequence-state line code .....	569
	Modified duobinary partial response coding .....	580
	Modulating a random process .....	263
	Modulation .....	19
	modulation .....	200
Multicarrier	modulation .....	255

NRZI modulation .....	262
Orthogonal multipulse modulation .....	306
Spread spectrum modulation .....	306
Multicarrier modulation (MCM) .....	307,693
Spread spectrum modulation .....	338
Amplitude and phase modulation, AM/PM .....	203
Amplitude-phase modulation (AM-PM) .....	219
Continuous-phase modulation (CPM) .....	243
Continuous-phase modulation (CPM) .....	589
Modulation index .....	589
Pulse Amplitude Modulation (PAM) .....	12
Pulse amplitude modulation (PAM) .....	182
Pulse-code modulation (PCM) .....	2,117
Quadrature-amplitude modulation (QAM) .....	20,202,218
Single-sideband modulation (SSB) .....	116
Modulo-two D-transform .....	593
Modulo-two summation .....	564
Characteristic or moment generating function .....	51
Moment generating function of shot noise .....	91
Monic minimum-phase spectral factorization .....	31,62
Monic polynomials and sequences .....	22
Monic transfer functions .....	24
Mean-square error (MSE) .....	467
Excess mean-square error (MSE) .....	536
Minimum mean-square error (MSE) estimator .....	521
MSE gradient algorithm (MSEG) .....	525,533,551
MSEG algorithm .....	526
Convergence of Minimum-shift keying (MSK) .....	244,589
Multicarrier modulation .....	255,307,693
Decision-directed equalization for multicarrier modulation .....	259
Multicarrier signaling and channel capacity .....	257
Multidimensional constellation .....	220,651,654
Multimode and single mode optical fiber .....	130
Multipath and selective fading .....	148
Delay spread for multipath channels .....	149
Frequency selective multipath fading .....	230
Multiple access .....	260,766
Code-division multiple access (CDMA) .....	261,766,787
Frequency-division multiple access (FDMA) .....	260,787
Time-division multiple access (TDMA) .....	774
Multiplexing .....	766
Frequency division multiplexing .....	116,256
Time-division multiplexing (TDM) .....	116,770
Statistical multiplexing .....	573,779
Frequency-division multiplexing (FDM) .....	117,167,797
Time-compression multiplexing (TCM) .....	797
Wavelength-division multiplexing (WDM) in optical fiber .....	789
Orthogonal multipulse modulation .....	230-260,306
Correlation receiver for orthogonal multipulse .....	232
Minimum bandwidth of orthogonal multipulse .....	235
Chang pulses for orthogonal multipulse .....	237
Discrete-time combined PAM and multipulse .....	253
PAM combined with orthogonal multipulse .....	284
Combined PAM and orthogonal multipulse signaling .....	249
Multitone data transmission .....	256

Vector or	multivariate Gaussian random variable .....	56
	Mutual information .....	98,103
	Narrowband channel model .....	149
	Natural or free-running frequency of a VCO .....	702
	Near- and far-end crosstalk (NEXT and FEXT) .....	127
Data or computer	network .....	4
Telecommunication	network .....	4
Telephone	network .....	4
Digital communications	network .....	765
Integrated services digital	network (ISDN) .....	4,119
Satellite	networks .....	766
Twoport	networks and chain matrix .....	123
Local and metropolitan area	networks (LAN and MAN) .....	4,5,766
Near- and far-end crosstalk	(NEXT and FEXT) .....	127
Thermal	noise .....	140
Thermal, quantization, and impulse	noise .....	162
C-notched and C-message	noise .....	163
Impulse	noise .....	257
Gaussian white	noise .....	61,104
Moment generating function of shot	noise .....	91
Granular	noise and zero-crossing jitter in magnetic .....	171
Equalization	noise enhancement .....	198,210,320,323,442,450,468
Signal and	noise generation models .....	379
Shot	noise or filtered Poisson process .....	424
Shot	noise or filtered Poisson process .....	83
Available	noise power .....	140
Signal to	noise ratio (SNR) .....	197,225
	Noise temperature .....	146
	Non-negative real transfer functions .....	29
Vector inner product and	norm .....	34
Gaussian or	normal random variables .....	53
	Normalized signal-to-noise ratio .....	350
Return-to-zero (RZ) and non-return-to-zero	(NRZ) pulse shapes .....	559
	NRZI modulation .....	262
	Nyquist criterion .....	190
Generalized	Nyquist criterion .....	235,266,304
	Nyquist echo canceler .....	808
	Nyquist sampling theorem and aliasing .....	16
Bytes and	octets .....	771
Bit and	octet interleaving .....	771
	Offered load of a queue .....	82
	Offset keyed PSK (OPSK or OK-PSK) .....	247
	Offset keyed QAM (OQAM or OK-QAM) .....	247
	One's density .....	605
	On-off keying (OOK) .....	261,361
	Optical fiber .....	128
Multimode and single mode	optical fiber .....	130
Receiver design for	optical fiber .....	261,360
Coherent	optical fiber .....	789
Wavelength-division multiplexing (WDM) in	optical fiber .....	789
	Optical fiber chromatic or material dispersion .....	131
	Optical fiber core, cladding, and sheath .....	129
	Optical fiber material attenuation .....	131
	Optical fiber mode dispersion .....	131
Coherent	optical fiber reception .....	365

Homodyne and heterodyne	optical fiber reception .....	365
Direct detection in	optical fiber systems .....	138
	Optical fiber waveguide .....	129
	Optical source linewidth .....	131
Magnetic and	optical storage .....	4
Quantum limit for fiber	optical transmission .....	361
Offset keyed QAM	(OQAM or OK-QAM) .....	247
	Orthogonal multipulse modulation .....	230-260,306
Correlation receiver for	orthogonal multipulse .....	232
Minimum bandwidth of	orthogonal multipulse .....	235
Chang pulses for	orthogonal multipulse .....	237
PAM combined with	orthogonal multipulse .....	284
Combined PAM and	orthogonal multipulse signaling .....	249
	Orthogonal vectors .....	35
	Orthogonality principle .....	524,533,551
Voltage-controlled	oscillator (VCO) .....	243,701,718
Random variable	outcome and sample space .....	49
	Packet field .....	778
	Packet or store-and-forward switching .....	777
	Packet speech .....	780
	Packet switch .....	782
Circuit and	packet switching .....	5
Link frame in	packet switching .....	778
Connection in	packet switching .....	779
Protocols in	packet switching .....	782
Synchronization to	packets .....	778
	Paley-Wiener condition .....	446
Pulse Amplitude Modulation	(PAM) .....	12,182
Baseband	PAM .....	191-198
Passband	PAM .....	199-212
Spectral efficiency of	PAM .....	223
Slicer design for	PAM .....	287
Discrete-time combined	PAM and multipulse .....	253
Combined	PAM and orthogonal multipulse signaling .....	249,284
	PAM pulse shape .....	181
	Parallel decision-feedback equalization .....	693
	Parity check matrix of convolutional codes .....	627
	Parity circuit .....	69
	Parity-check codes .....	613
	Parity-check matrix .....	622
	Parseval's relationships .....	44
	Partial response .....	413
Dicode	partial response .....	579
Duobinary	partial response .....	579,584,740
Precoding in	partial response .....	581
Dicode	partial response .....	583
Modified duobinary	partial response coding .....	580
	Partial-response CPM .....	590
Lattice	partition .....	685
Mapping by set	partitioning .....	676,688
	Passband adaptive equalization .....	546
	Passband echo canceler .....	804
Analytic	passband filter .....	204
	Passband PAM .....	199-212
Canonical representation of	passband signal .....	19

	Passband spectral-line timing recovery .....	747
	Passband transversal filter .....	807
Branch and	path metric for the Viterbi algorithm .....	414
Trellis diagram nodes, branches, and	paths .....	413
Survivor and merged	paths for the Viterbi algorithm .....	416
Pulse-code modulation	(PCM) .....	2,117
	Peak deviation in FSK .....	242,244
	Peak- or average-power constraint .....	118
PLL	peaking .....	707
	Perfect codes .....	620
Timing	phase .....	739,749
Propagation, attenuation, and	phase constants .....	120
	Phase detector .....	701
Sawtooth	phase detector .....	702
Sinusoidal	phase detector .....	713
Triangular	phase detector .....	717
Quadrature phase lock of sinusoidal	phase detectors .....	714
Sampling	phase detectors .....	715
Exclusive-or	phase detectors .....	716
Adaptive	phase detectors .....	718
	Phase diagrams for CPM .....	590
Random	phase epoch .....	66
	Phase jitter .....	726
Quadrature	phase lock of sinusoidal phase detectors .....	714
Amplitude and	phase modulation, AM/PM .....	203
	Phase response .....	15
	Phase shift filter .....	44
	Phase splitter .....	18
	Phase-locked loop (PLL) .....	700
	Phase-shift keying (PSK) .....	203,214,219
Binary	phase-shift keying (BPSK) .....	214,219
Quadrature	phase-shift keying (QPSK) .....	214,219
	Photodetector dark current .....	137
Photodetectors, PIN diode and avalanche	Photodetectors, PIN diode and avalanche .....	136
	photodiode .....	136
	Physical layer .....	7
	Physical media and channels .....	115
Photodetectors,	PIN diode and avalanche photodiode .....	136
Phase-locked loop	(PLL) .....	700
Lock or hold-in range of a	PLL .....	704
Order of a	PLL .....	705
Type of a	PLL .....	708,713
Discrete-time	PLL .....	709
	PLL clicks .....	702
	PLL peaking .....	707
Capture of	PLLs .....	709
Pull-in time of	PLLs .....	709
Seize range of	PLLs .....	709
Digital	PLLs .....	715
Mixed discrete and continuous-time	PLLs .....	715
	Poisson distribution and process .....	78,79
Shot noise or filtered	Poisson process .....	424
Pure birth or	Poisson process .....	78
Shot noise or filtered	Poisson process .....	83
Time-varying	Poisson process .....	90

	Poisson process and queueing .....	75
	Poisson's sum formula .....	759
Conjugate-reciprocal	pole-pole pairs .....	30
	Poles .....	24
Conjugate-reciprocal	pole-zero pairs .....	26
Roll-call	polling .....	781,783
Token-passing	polling .....	783
Generator	polynomial .....	593,624
Irreducible	polynomials .....	598
Primitive	polynomials .....	600
Monic	polynomials and sequences .....	22
	Positive definite matrix .....	522,526
DFE	postcursor equalizer .....	456
Precursor and	postcursor ISI .....	212,454
	Posterior probability .....	381
Link	power budget .....	144
	Power control in CDMA .....	791
Average	power of a deterministic signal .....	13
	Power spectral density of a random process .....	58,551
Transmit	power spectrum .....	187
	Power spectrum of a cyclostationary process .....	86
	Power spectrum of a Markov chain .....	87
Transmitter	precoding .....	460,574,689
Differential encoding or	precoding .....	564
Flexible	precoding .....	694
	Precoding in partial response .....	581
	Precursor and postcursor ISI .....	212,454
	Precursor equalizer .....	299,335,455
Linear	prediction .....	62,459,473
	Prefiltering in timing recovery .....	746,748
	Primary and secondary parameters of cable .....	124
	Primitive polynomials .....	600
	Prior or <i>a priori</i> probabilities .....	381
State transition	probabilities of a Markov chain .....	70
Posterior	probability .....	381
Sequence error	probability .....	418
Error events w.s. bit and symbol error	probability .....	432
Conditional	probability .....	52
Marginal	probability .....	52
	Probability density function .....	49
Complementary	probability distribution function .....	53
Maximum a-posteriori	probability (MAP) .....	379
	Probability of an event .....	49
	Probability of block error, block error rate .....	328
	Probability of error .....	3,382
Union bound on the	probability of error .....	327
	Probability of error for BSC .....	389
Stochastic or random	processes .....	48,57
Propagation and	processing delay .....	3
	Processing gain in spread spectrum .....	343
	Projection theorem .....	37
DFE error	propagation .....	456,460,511
	Propagation and processing delay .....	3
	Propagation, attenuation, and phase constants .....	120,145
Error	propagation in differential encoding .....	729

	Proportional plus integral loop filter .....	708
	Protocols in packet switching .....	782
	Pseudorandom sequences .....	592,601
	Pseudoternary line codes .....	560
Phase-shift keying (PSK) .....		203,214,219
Differential PSK .....		221
differential binary PSK (DBPSK) .....		221
Differential PSK (DPSK) .....		221
Offset keyed PSK (OPSK or OK-PSK) .....		247
Quadrature PSK (QPSK) .....		221
	Pull-in time of PLLs .....	709
Equivalent baseband pulse .....		208
Isolated pulse .....		225
	Pulse amplitude modulation (PAM) .....	12,182
	Pulse code modulation (PCM) .....	2
PAM pulse shape .....		181
Biphase or Manchester pulse shape .....		559
Wal2 pulse shape .....		604
(RZ) and non-return-to-zero (NRZ) pulse shapes .....		559
	Pulse-code modulation (PCM) .....	117
Raised-cosine pulses .....		190
Roll-off factor of raised-cosine pulses .....		190
Square-root raised cosine pulses .....		228
Self equalizing pulses .....		560
Chang pulses for orthogonal multipulse .....		237
	Pure and slotted ALOHA .....	785
	Pure birth or Poisson process .....	78
	Q() function, tail of a Gaussian distribution .....	53,389
Quadrature amplitude modulation (QAM) .....		20,202,218,219
Offset keyed QAM (OQAM or OK-QAM) .....		247
Quadrature phase-shift keying (QPSK) .....		214,219,221
	Quadrature amplitude modulation (QAM) .....	10,202,218,219
In-phase and quadrature component .....		202
	Quadrature phase lock of sinusoidal phase .....	714
	Quadrature phase-shift keying (QPSK) .....	214,219,221
Thermal, quantization, and impulse noise .....		162
	Quantized feedback .....	592
	Quantizer, slicer, decision .....	184
	Quantum efficiency and responsivity .....	136
	Quantum limit for fiber optical transmission .....	361
	Quasiperfect codes .....	621
Offered load of a queue .....		82
	Queue management disciplines .....	781
Poisson process and queueing .....		75
	Queueing, buffers, waiting positions, and .....	80
	Queueing delay and waiting time .....	82,780
Cellular mobile radio .....		142,793
Digital radio .....		142
Microwave radio .....		142,788
IS-54 standard for digital cellular radio .....		221
Radio system margin and rain attenuation .....		148
	Raised-cosine pulses .....	190
Square-root raised cosine pulses .....		228
Roll-off factor of raised-cosine pulses .....		190
	Random or stochastic processes .....	57



	Random phase epoch .....	66
Stochastic or	random process .....	48
Modulating a	random process .....	263
Gaussian	random process .....	57
Mean and autocorrelation of a	random process .....	57
Strict sense stationary	random process .....	57
Autocorrelation function of a	random process .....	58
Power spectral density of a	random process .....	58
Wide sense stationary (WSS)	random process .....	58
White	random process .....	59
Cross-correlation function of a	random process .....	61
Cross-spectral density of a	random process .....	61
Linear prediction of a	random process .....	62
Cyclostationary	random process .....	65,741
Cumulative distribution function of a	random variable .....	49
Standard Gaussian	random variable .....	53
Vector or multivariate Gaussian	random variable .....	56
	Random variable outcome and sample space .....	49
Joint distributions of	random variables .....	50
Cross-correlation of	random variables .....	51
Uncorrelated	random variables .....	51
Gaussian or normal	random variables .....	53
Jointly Gaussian	random variables .....	55
Source coding and	rate distortion .....	98
	Rate of a source .....	101
	Rate-normalized SNR .....	347,350,501
	Rational transfer functions .....	24
	Rayleigh fading .....	152
	Rayleigh scattering .....	132
Running digital sum	(RDS) .....	558,575
	Real and complex LTI systems .....	13
Non-negative	real transfer functions .....	29
Correlation and matched filter	receiver .....	225,252,289
Discrete-time correlation or matched-filter	receiver .....	255,257
	Receiver design for optical fiber .....	261,360
	receiver for orthogonal multipulse .....	232
Coherent and incoherent	receivers .....	240,247
Digital magnetic	recording .....	167,262
A.c.-bias and saturation magnetic	recording .....	169
noise and zero-crossing jitter in magnetic	recording .....	171
Timing	recovery .....	183,211,737-764
Carrier	recovery .....	211,725-736
Framing	recovery .....	771
	Rectangular constellations .....	219
	Reduced-state sequence detection (RSSD) .....	692
	Redundancy .....	181,557
	Reed-Solomon codes .....	624
	Reflected transfer function .....	29
Source and	reflected waves .....	120
Total internal	reflection .....	128
Voltage	reflection coefficient .....	122
	Regeneration principle .....	6,168,378
	Regenerative repeaters .....	118
	Region of convergence (ROC) .....	22
Regenerative effect and regenerative	repeater .....	6,118

	Repetition codes .....	645
Coset	representative .....	685
Quantum efficiency and	responsivity .....	136
	Return-to-zero (RZ) and non-return-to-zero .....	559
	Right-sided and left-sided sequences .....	21
	Rings .....	637
Slotted	ring .....	774
Token-passing	ring .....	784
Bus,	ring, and tree topologies .....	768
Region of convergence	(ROC) .....	22
	Roll-call polling .....	781,783
	Roll-off factor of raised-cosine pulses .....	190
	Rotated data symbols .....	547,805
Reduced-state sequence detection	(RSSD) .....	692
	Running digital sum (RDS) .....	558,575
Return-to-zero	(RZ) and non-return-to-zero (NRZ) pulse shapes .....	559
Random variable outcome and	sample space .....	49
	Sampled matched filter .....	295
	Sample-derivative timing recovery .....	753
	Sampling interval and frequency .....	11
	Sampling phase detectors .....	715
	Sampling theorem .....	16
Nyquist	sampling theorem and aliasing .....	16
	Satellite channels .....	142,240
	Satellite networks .....	766
A.c.-bias and	saturation magnetic recording .....	169
	Sawtooth phase detector .....	702
Rayleigh	scattering .....	132
	Schwarz inequality .....	38
Integral form of the	Schwarz inequality .....	226
Vector form of the	Schwarz inequality .....	38
Cryptographic or frame-synchronized	scrambler .....	592,593
	Seize range of PLLs .....	709
Frequency	selective multipath fading .....	148,230
	Self- and frame-synchronized scramblers .....	592
	Self equalizing pulses .....	560
Average	self-information or entropy .....	99
	Self-timing .....	737
Light-emitting diode and	semiconductor laser .....	135
MAP	sequence detector .....	430
Maximum-likelihood	sequence detector (MLSD) .....	406,409,431,442,448,480
Error events in the ML	sequence detector .....	418
Reduced-state	sequence detection (RSSD) .....	692
	Sequence error probability .....	418
	Sequence-state line code .....	566
Modes of a	sequence-state line code .....	569
Queueing, buffers, waiting positions, and	servers .....	80
Mapping by	set partitioning .....	676,688
Stochastic gradient	(SG) algorithm .....	532,812
Convergence of	SG algorithm .....	535
	Shadowing effect .....	152
	Shannon information theory .....	97
	Shaping and coding gain .....	652
	Shaping by shell mapping .....	694
Optical fiber core, cladding, and	sheath .....	129

Shaping by	shell mapping .....	694
Maximum-length	shift register (MLSR) .....	591,599
Feedback	shift registers .....	592
Markov chains and	shift-register process .....	410,630
	Shot noise or filtered Poisson process .....	83,424
Moment generating function of	shot noise .....	91
	Signal and noise generation models .....	379
Definition of	signal constellation .....	213
Binary antipodal	signal constellation .....	214,324
Multidimensional	signal constellation .....	654
	Signal flow graphs .....	71
	Signal space .....	31
Geometric structure of	signal space .....	34
	Signal to noise ratio (SNR) .....	197,225
Complex-valued	signals .....	12
Anticausal	signals .....	273
Discrete-time analytic	signals .....	273
	Signal-space codes .....	610,650
	Signal-to-noise ratio (SNR) .....	107
Normalized	signal-to-noise ratio .....	350
Multimode and	single mode optical fiber .....	130
	Single-sideband modulation (SSB) .....	116
	Singular autocorrelation matrix .....	544
	Sinusoidal phase detector .....	713
Quadrature phase lock of	sinusoidal phase detectors .....	714
	Skin effect .....	125
Quantizer,	slicer, decision .....	184,213
	Slicer decision regions .....	215,288
	Slicer design for PAM .....	287
	Slope distortion .....	161
Pure and	slotted ALOHA .....	785
	Slotted ring .....	774
Signal-to-noise ratio	(SNR) .....	107,197,225
Matched filter bound on the	SNR .....	226
Rate-normalized	SNR .....	347,350,501
Hard and	soft decoding .....	610,650
	Solitons .....	141
	Source and channel coding .....	98
	Source and reflected waves .....	120
	Source coding and rate distortion .....	98
	Source coding theorem .....	101
Signal	space .....	31
Euclidean	space .....	32
Inner product and Hilbert	space .....	36
	Space- and time-division switching .....	774
Matrix	spectral decomposition .....	527
Power	spectral density of a random process .....	58
	Spectral efficiency .....	185,304,349
	Spectral efficiency of PAM .....	223
	Spectral emission masks .....	147
Monic minimum-phase	spectral factorization .....	31,62,458
	Spectral nulls using filtering .....	574
	Spectral-line timing recovery method .....	741
Passband	spectral-line timing recovery .....	747
Folded	spectrum .....	228,290

Transmit power	spectrum	187
Power	spectrum of a cyclostationary process	86
Power	spectrum of a Markov chain	87
Packet	speech	780
	Spread spectrum	229,306,338,357
Chip waveform in	spread spectrum	339
Spreading code in	spread spectrum	339,374
Direct-sequence	spread spectrum	341,790
Processing gain in	spread spectrum	343
	Square-law timing recovery	743
	Square-root raised cosine pulses	228
Single-sideband modulation	(SSB)	116
BIBO	stability	23
	Stable sequences	23
	Standard Gaussian random variable	53
CCITT	standards	166,240,373
	Stationary Markov chain	71
Strict sense	stationary random process	57
Wide sense	stationary (WSS) random process	58
	Statistical independence	50
	Statistical multiplexing	573,779
Sufficient	statistics	397
Unit	step function	45
	Step size normalization	539
	Step size of gradient algorithm	526
	Stochastic gradient (SG) algorithm	532,812
	Stochastic gradient algorithm in timing	749
	Stochastic or random processes	48,57
Magnetic and optical	storage	4
Packet or	store-and-forward switching	777
	Strict sense stationary random process	57
	Strictly and loosely maximum-phase transfer	27
Digital	subscriber loop	119,263,769,789,797
	Subspace of a linear space	37
	Sufficient statistics	397
Running digital	sum (RDS)	558,575
	Superframe	772
Echo	suppressors and cancelers	166
	Survivor and merged paths for the Viterbi	416
	Switching	4
Circuit and packet	switching	5,773,782
Space- and time-division	switching	774
Packet or store-and-forward	switching	777
	symbol alphabet	213
	Symbol error	324
Error events vs. bit and	symbol error probability	432
	Symbol period, rate	181,188
Data	symbols and alphabet	181,380
Rotated data	symbols	547,805
Circularly	symmetric Gaussian process	313
Coherent	(synchrondyne) and differential detection	222
Definition of	synchronization	700
Frame	synchronization	594
	Synchronization to packets	778
	Syndrome	623

	Systematic codes .....	614
	Systematic convolutional coders .....	626
	Systematic or data-dependent timing jitter .....	757
Linear time-invariant (LTI)	systems .....	13
Real and complex LTI	systems .....	13
Causal and anti-causal	systems .....	22
	T1-carrier system .....	3
QO function,	tail of a Gaussian distribution .....	53
	Talker and listener echo .....	166
	Tap weights or coefficients .....	487
Bridged	taps .....	172
Time-compression multiplexing	(TCM) .....	797
Time-division multiplexing	(TDM) .....	770
	TDM frame and framing bits .....	771
Time-division multiple access	(TDMA) .....	774
Demand assignment in	TDMA .....	775
	Telecommunication network .....	4
Digital cellular	telephone .....	792
	Telephone channels .....	160
	Telephone network .....	4
Noise	temperature .....	146
Asymptotic equipartition	theorem .....	100
Source coding	theorem .....	101
Asymptotic equipartition	theorem .....	110
Sampling	theorem .....	16
Central limit	theorem .....	250
Landau-Pollak	theorem .....	305
Projection	theorem .....	37
Bayes'	theorem .....	53
Central limit	theorem .....	55
Campbell's	theorem .....	84
Channel capacity and channel coding	theorem .....	98
Nyquist sampling	theorem and aliasing .....	16
Final value	theorem for Laplace transforms .....	708
Final value	theorem for Z transforms .....	712
Fundamental	theorem of expectation .....	50
	Thermal, quantization, and impulse noise .....	140,162
	Time- and frequency-division multiplexing .....	116
	Time-compression multiplexing (TCM) .....	797
	Time-division multiple access (TDMA) .....	774
	Time-division multiplexing (TDM) .....	770
Space- and	time-division switching .....	774
Linear	time-invariant (LTI) systems .....	13
	Time-varying Poisson process .....	90
Envelope derived	timing .....	747
	Timing function .....	754
	Timing jitter .....	3,738
Accumulation of	timing jitter .....	757
Systematic or data-dependent	timing jitter .....	757
	Timing phase .....	739,749
	Timing recovery .....	183,211,737-764
Deductive and inductive	timing recovery .....	738
Square-law	timing recovery .....	743
Absolute-value	timing recovery .....	745
Fourth-power	timing recovery .....	745

Prefiltering in	timing recovery .....	746
Passband spectral-line	timing recovery .....	747
LMS	timing recovery .....	748
ML	timing recovery .....	748
MMSE	timing recovery .....	748
Prefiltering for	timing recovery .....	748
Stochastic gradient algorithm in	timing recovery .....	749
Decision-directed	timing recovery .....	750
Sample-derivative	timing recovery .....	753
Baud-rate	timing recovery .....	754
Spectral-line	timing recovery method .....	741
	Timing recovery performance .....	739
	Toeplitz matrix .....	522, 531
	Token-passing polling .....	783
	Token-passing ring .....	784
	Tomlinson-Harashima coding .....	460
Bus, ring, and tree	topologies .....	768
Adaptation	training signal .....	519
Reflected	transfer function .....	29
	Transfer functions .....	14
Monic	transfer functions .....	24
Rational	transfer functions .....	24
Allpass	transfer functions .....	25
Strictly and loosely minimum-phase	transfer functions .....	27
Non-negative real	transfer functions .....	29
Fourier	transform .....	13
Discrete-time Fourier	transform .....	14
Z	transform .....	21
Discrete-time Hilbert	transform .....	273
Balanced transmission and	transformer coupling .....	127
Hilbert	transforms in the frequency domain .....	273
State	transition diagram of a Markov chain .....	69
State	transition probabilities of a Markov chain .....	70
Friis	transmission equation .....	144
Uniform	transmission line .....	119
	Transmit power spectrum .....	187
	Transmitter precoding .....	460, 574, 689
	Transversal filter .....	487
Finite	transversal filter .....	518
Baseband and passband	transversal filter .....	520, 807
LMS	transversal filter algorithm .....	532
Bus, ring, and	tree topologies .....	768
	Trellis coding .....	668
Set partitioning for	trellis codes .....	676
	Trellis diagram nodes, branches, and paths .....	413
	Triangular phase detector .....	717
	Tributary bit-streams .....	770
	Truncation depth in the Viterbi algorithm .....	417
	Twinned binary line code .....	561
	Twisted pair and coaxial cable .....	119
Hybrid for	two- to four-wire conversion .....	165
	Twoport networks and chain matrix .....	123
	Type of a PLL .....	708
	Type of a discrete-time PLL .....	713
	Unbiased DFE-MSE (DFE-MSE-U) .....	479

	Uncorrelated random variables .....	51
	Uniform transmission line .....	119
	Union bound .....	49,318,
	Union bound on the probability of error .....	327
Double zeros on the	unit circle .....	30
	Unit step function .....	45
	Unitary matrices .....	268
	Variable-rate coding .....	778
Mean and	variance .....	50
Digital sum	variation (DSV) .....	562
Voltage-controlled oscillator	(VCO) .....	243,701,718
Natural or free-running frequency of a	VCO .....	702
	Vector form of the Schwarz inequality .....	38
Capacity of	vector Gaussian noise channel .....	108
	Vector inner product and norm .....	34
	Vector or multivariate Gaussian random variable .....	56
Detection of a	vector signal .....	385
Linear space or	vector space .....	31
	Vector space over Galois fields .....	639
Orthogonal	vectors .....	35
MAP detector for	vectors .....	388
Voice-frequency	(VF) channel .....	116
Bipolar	violation .....	604
	Viterbi algorithm .....	409
Branch and path metric for the	Viterbi algorithm .....	414
Survivor and merged paths for the	Viterbi algorithm .....	416
Truncation depth in the	Viterbi algorithm .....	417
	Viterbi algorithm applied to convolutional .....	631
	Viterbi detector .....	198
Attenuation of	voiceband channel .....	162
	Voiceband data modem .....	186,769
	Voice-frequency (VF) channel .....	116
	Voltage-controlled oscillator (VCO) .....	243,701,718
Fundamental	volume of a lattice .....	656
Queueing, buffers,	waiting positions, and servers .....	80
Queueing delay and	waiting time .....	82
	Wal2 pulse shape .....	604
Baseline	wander in baseband PAM systems .....	557
	Water-pouring spectrum for channel capacity .....	491
Optical fiber	waveguide .....	129
	Wavelength .....	120
Wavelength-division multiplexing	(WDM) in optical fiber .....	789
Gaussian	white noise .....	61
	White random process .....	59
	Whitened matched filter (WMF) .....	406,442,445
	Whitening filter .....	62,392
	Wide area networks .....	5
	Wide sense stationary (WSS) random process .....	58
Whitened matched filter	(WMF) .....	406,442,445
Wide sense stationary	(WSS) random process .....	58
HDLC and	X.25 .....	778
	Z transform .....	21
Final value theorem for	Z transforms .....	712
Granular noise and	zero-crossing jitter in magnetic recording .....	171
	Zero-disparity line code .....	571

	Zero-forcing criterion (ZF) .....	189
	Zero-forcing decision-feedback equalizer .....	453,475,691
	Zero-forcing linear equalization (LE-ZF) .....	451,469
	Zeros .....	24
Double	zeros on the unit circle .....	30
Conjugate-reciprocal	zero-zero pairs .....	30
Zero-forcing criterion	(ZF) .....	189